Department of Micro and Nanosciences

All-Digital Phase-Locked Loop for Radio Frequency Synthesis

Liangge Xu

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 46/2014

All-Digital Phase-Locked Loop for Radio Frequency Synthesis

Liangge Xu

A doctoral dissertation completed for the degree of Doctor of Science in Technology to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall S1 of the school on 23 of May 2014 at 12:00.

Aalto University
School of Electrical Engineering
Department of Micro and Nanosciences

Supervising professor
Professor Jussi Ryynänen

Thesis advisor
Dr. Kari Stadius

Preliminary examiners
Professor Howard Luong, Hong Kong University of Science & Technology, China
Professor Pietro Andreani, Lund University, Sweden

Opponent
Professor Stefan Heinen, RWTH Aachen University, Germany

Aalto University publication series
DOCTORAL DISSERTATIONS 46/2014

ISBN 978-952-60-5637-1
ISBN 978-952-60-5638-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5638-8

Unigrafia Oy
Helsinki 2014

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Liangge Xu

Name of the doctoral dissertation
All-Digital Phase-Locked Loop for Radio Frequency Synthesis

Publisher

Unit Department of Micro and Nanosciences

Series Aalto University publication series DOCTORAL DISSERTATIONS 46/2014

Field of research Electronic Circuit Design

Manuscript submitted 3 December 2013 **Date of the defence** 23 May 2014

Permission to publish granted (date) 25 March 2014 **Language** English

☒ **Monograph** ☐ **Article dissertation (summary + original articles)**

Abstract

It has been a constant challenge in wireless system design to meet the growing demand for an ever higher data rate and more diversified functionality at minimal cost and power consumption. The key lies in exploiting the phenomenal success of CMOS technology scaling for high-level integration. This underlies the paradigm shift in the field of integrated circuit (IC) design to one that increasingly favours digital circuits as opposed to their analog counterparts. With radio transceiver design for wireless systems in particular, a noticeable trend is the introduction of digital-intensive solutions for traditional analog functions. A prominent example is the emergence of the all-digital phase-locked loop (ADPLL) architectures for frequency synthesis. By avoiding traditional analog blocks, the ADPLL brings the benefits of high-level integration and improved programmability.

This thesis presents ADPLL frequency synthesizer design, highlighting practical design considerations and technical innovations. Three prototype designs using a 65-nm CMOS technology are presented. The first example address a low-power ADPLL design for 2.4-GHz ISM (Industrial, Scientific, Medical) band frequency synthesis. A high-speed topology is employed in the implementation for the variable phase accumulator to count full cycles of the radio frequency (RF) output. A simple technique based on a short delay line in the reference signal path allows the time-to-digital converter (TDC) core to operate at a low duty cycle with approximately 95% reduction in its average power consumption. The ADPLL incorporates a two-point modulation scheme with an adaptive gain calibration to allow for direct frequency modulation. The second implementation is a wide-band ADPLL-based frequency synthesizer for cognitive radio sensor units. It employs a digitally controlled ring oscillator with an LC tank introduced to extend the tuning range and reduce power dissipation. An adaptive frequency calibration technique based on binary search is used for fast frequency settling. The third implementation is another wideband ADPLL frequency synthesizer. At the architectural level, separation of coarse-tune and fine-tune branches results in a word length reduction for both of them and allows the coarse tuning logic to be powered off or clock gated during normal operation, which led to a significant reduction in the area and power consumption for the digital logic and simplified the digital design. A dynamic binary search technique was proposed to achieve further improved frequency calibration speed compared with previous techniques. In addition, an original technique was employed for the frequency tuning of the wideband ring oscillator to allow for compact design and excellent linearity.

Keywords radio frequency, all-digital phase-locked loop, digitally controlled oscillator, time-to-digital converter, frequency synthesizer

ISBN (printed) 978-952-60-5637-1 **ISBN (pdf)** 978-952-60-5638-8

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2014

Pages 173 **urn** http://urn.fi/URN:ISBN:978-952-60-5638-8

Preface

The research work for this thesis was carried out in the Electronic Circuit Design Laboratory (ECDL) between 2007 and 2011. During the research work, I received numerous help and support from a lot of people. Without them, the work would not have been accomplished.

I would like to thank my supervisor Professor Jussi Ryynänen for his guidance, encouragement and support throughout my doctoral study on this interesting subject. My sincere thanks also go to my instructor Dr. Kari Stadius for his tremendous effort and time to instruct me on my research work.

I am grateful to Professor Saska Lindfors, for giving me the privileged opportunity to start my work on the project and guiding me through my diploma study. I would like to acknowledge and thank all my former colleagues in the ECDL for their support and useful discussions.

The work has received funding from the Finnish Funding Agency for Technology and Innovation (TEKES). In addition, the research leading to these results has received funding from the European Community's Seventh Framework Programme. Their support are greatly appreciated.

Finally, I thank my parents for their unimaginable sacrifice that allowed me to pursue quality education.

Espoo, March 21, 2014,

Liangge Xu

Contents

List of symbols

α, ρ	DLPF parameters
$\Delta\phi$	Phase value represented by a digital control word
$\Delta\theta_R$	Excess phase fluctuation of the frequency reference signal
Δf	Frequency resolution of an ADPLL
Δf	Offset frequency
Δf_{dco}	DCO frequency resolution
Δt	time difference
Δt_f	Time interval starting from a falling edge
Δt_r	Time interval starting from a rising edge
Δt_{tdc}	TDC time resolution
Δt_R	Absolute timing jitter of the reference clock
$\Delta t'_r$	Time interval ending at a rising edge
ΔT	Time interval
ω_{dco}	Angular frequency of a DCO
ω_{ON}	Frequency of a ring LC oscillator with the LC tank connected
ϕ	Carrier phase error
$\phi(t)$	Phase as a function of time
Φ_{e}	Digital signal of the detected phase error
Φ_{v}	Digital signal of the detected phase
ψ	A real-valued variable
σ_ϕ	Root mean square phase error
σ_j	RMS jitter
τ	Delay of each stage
τ_1, τ_2	Delay line unit delays in Vernier-line TDC
τ_{d}	Delay
τ_{f}	Delay of a unit cell for a falling edge
τ_{inv}	Interter delay
τ_{lsb}	Delay of a unit cell corresponding to one LSB TDC output

τ_{off}	Delay offset
τ_{r}	Delay of a unit cell for a rising edge
θ_{dco}	Phase of a DCO output
θ_R	Reference phase
ε	Normalized TDC output
$a_0 ... a_{\mathrm{n}}$	Control code values
a_i, b_j	Coefficients of a generic loop filter
A_{st}	Magnitude of step change in FCW
$A(z)$	Gain of the forward path in an ADPLL
A	Silicon area
A	Amplitude
$B(z)$	Gain of the feedback path in an ADPLL
C	Capacitance
C_1, C_2	Capacitor values in a loop filter
C_L	Load capacitance
C_p	Capacitance in an LC tank for a ring LC oscillator
d	DCO control code
d_x	DCO control code
D	DCO control code from the DLPF output in an ADPLL
D_{cnt}	Counter output
D_{dline}	Delay line output
D_{I}	DLPF integral branch output
D_{out}	TDC output
D_P	DLPF proportional branch output
e_{q}	Input-referred quantization error of a TDC
E	Magnitude of an error vector
E_{q}	Output quantization error of a TDC or PDC
$E(z)$	z-transform of an error signal
f	Frequency
f_0	Carrier frequency
f_0	DCO frequency with a zero control code
f_0	VCO frequency with a zero control voltage
f_0, f_1, f_2	Frequency constants
f_{clk}	Clock frequency
f_{dco}	DCO output frequency
f_{dith}	Dithering frequency
f_{in}	Frequency of the input signal
f_{lsb}	One-LSB frequency step

f_{LO}	Frequency of the LO signal
f_m	Offset frequency
f_{max}	Maximum frequency
f_{min}	Minimum frequency
f_{out}	Output frequency of a VCO or DCO
f_{osc}	Oscillation frequency
f_{ref}	Reference frequency
f_{res}	DCO frequency resolution
f_R	Reference frequency
f_{RF}	Frequency of the RF signal
f_s	Sampling frequency
f_x	DCO frequency for a specific code value
F_{div}	Frequency division ratio
$F_{\mathrm{div,F}}$	Fractional part of the frequency division ratio
$F_{\mathrm{div,I}}$	Integer part of the frequency division ratio
F_e	Digital signal of the detected frequency error
F_{ea}	Accumulated frequency error
F_{tar}	Target frequency ratio represented by FCW for an ADPLL
F_v	Digital signal of the detected frequency
$G_\Phi(f)$	An ADPLL transfer function
$G_{f,N}(z)$	Noramlized ADPLL overall transfer function
$G_f(z)$	ADPLL overall transfer function in the z-transform domain
G_{lpf}	DC gain of a loop filter
G_m	Transconductance of one stage in a ring oscillator
$G_{m,i}$	The ith-stage transconductance
$G(j\omega)$	Frequency response of a gain block in an oscillator model
$h_{\mathrm{zoh}}(t)$	Impulse response of the zero-order holding function
$H(j\omega)$	Filter frequency response in an oscillator model
$H(j\omega)$	Open loop transfer function of a oscillator model
$H_{\mathrm{zoh}}(f)$	Frequency response of the zero-order holding function
$H_{lp}(z)$	Transfer function of a low-pass filter
$H_{out}(f)$	Output transfer function in an ADPLL
$H(z)$	Loop filter transfer function
I	Current
I_{DD}	Charging/discharging current in a ring oscillator
I_{in}	Input current
I_p, I_n	Charging and discharging currents from a charge pump
$\Im$	Imaginary part (of an operand)

K_{dco}	DCO gain
K_{vco}	VCO gain
K_I	Gain of the integrator branch in a loop filter
K_P	Gain of the proportional branch in a loop filter
k	Sample index
L	Number of delay cells
L	Inductance
L_p	Inductance in an LC tank for a ring LC oscillator
$\mathcal{L}_{DJ}$	Phase noise level contributed by of delay line jitter
$\mathcal{L}\{f\}$	Phase noise level at offset frequency f
$\mathcal{L}_{int}$	Integrated phase noise
$\mathcal{L}_{QTDC}$	Phase noise level contributed by the TDC quantization
$\mathcal{L}$	Phase noise level
M	Number of bits
M	TDC output value as number of taps
n	Number of unit cells
n	Order of an SDM
N	Number of stages of a ring oscillator
P	Power consumption
P_{total}	Total power of a signal
R_1	Resistor in a loop filter
R_i	Load resistance of an ith stage of a ring oscillator model
R_L	Load resistance
$\Re$	Real part (of an operand)
r	Circle radius
r	Slope of a rising voltage
t	Time
t_s	Sampling time
T_0	Carrier period
T_{ckv}	Period of the RF signal to the TDC
T_{dco}	Period of DCO output
T_{in}	Input time interval
T_{lsb}	Input change corresponding to one LSB in the TDC output
T_{res}	TDC time resolution
T_R	Period of the frequency reference
T_v	Period of the RF signal to a TDC
$\overline{T_v}$	Averaged or low-passed filtered value of T_v
V_0	Voltage constant

V_a	Voltage at the input of a unit cell
V_b	Voltage at the output of a unit cell
V_{ctrl}	Control voltage of a VCO
V_i	Vottage tapped at interpolation point
$V_{in}(j\omega)$	Input voltage in the frequency domain
V_{osc}	Voltage amplitude of oscillation
$V_{out}(j\omega)$	Output voltage in the frequency domain
W	Word length
$W_{_F}$	Number of fractional bits
$x_{_{LO}}(t)$	Waveform of the LO signal as a function of time
$X_i(f)$	Spectrum of the input signal
$X_{_{LO}}(f)$	Spectrum of the LO signal
$X_m(f)$	Spectrum of the mixing result
$X_{\text{ref}}(f)$	Spectrum of the LO signal
$X(z)$	z-transform of an input signal
$Y(z)$	z-transform of an output signal

List of abbreviations

AC	Alternating current
ADC	Analog-to-digital converter
ADPLL	All digital phase-locked loop
AFC	Adaptive frequency calibration
A/D	Analog-to-digital
BER	Bit error rate
BS	Binary search
CMOS	Complimentary metal oxide semiconductor
DAC	Digital-to-analog converter
DAS	Direct analog frequency synthesis
DC	Direct current
DDS	Direct digital frequency synthesis
DECT	Digital Enhanced Cordless Telecommunications
DEM	Dynamic element matching
DLL	Delay-locked loop
DLPF	Digital loop filter
DNL	Differential nonlinearity
DPLL	Digital phase-locked loop
DSDM	Digital sigma-delta modulator
DWA	Data-weighted averaging
D/A	Digital-to-analog
D/F	Digital-to-frequency
EVM	Error vector magnitude
FCW	Frequency control word
FDC	Frequency-to-digital converter
FEA	Frequency error accumulator
FED	Frequency error detector
FIR	Fininte impulse response

FOM	Figure of merit
FSM	Finite state machine
ENOB	Effective number of bits
GPRS	General packet radio service
GRO	Gated ring oscillator
GSM	Global system for mobile communications
I	In-phase
IC	Integrated circuit
IF	Intermediate frequency
IIR	Infinite impulse reponse
INL	Integral nonlinearity
ISM	Industrial, scientific and medical
LNA	Low noise amplifier
LO	Local oscillator
LPF	Low-pass filter
LMS	Least mean square
LSB	Least significant bit
MASH	Multi-stage noise shaping
MMD	Multi-modulus frequency divider
MSB	Most significant bit
NCO	Numerically controlled oscillator
PA	Power amplifier
PAC	Phase-to-amplitude converter
PDC	Phase-to-digital converter
PFD	Phase frequency detector
PI	Proportional-integral
PLL	Phase-locked loop
PSD	Power spectral density
PVT	Process-voltage-temperature
Q	Quadrature phase, quality factor
QPSK	Quadrature phase shift keying
RF	Radio frequency
RMS	Root mean square
ROM	Read-only memory
RX	Receiver
SDM	Sigma-delta modulator
SNR	Signal-to-noise ratio
SNDR	Signal-to-distortion-and-noise ratio

SSB	Single sideband
TA	Time amplifier
TAC	Time-to-analog converter
TDC	Time-to-digital converter
TX	Transmitter
UWB	Ultra-wide band
VCDL	Voltage-controlled delay line
VCO	Voltage controlled oscillator
VGA	Variable gain amplifier
VPA	Variable phase accumulator
VPAC	Variable phase accumulator
WCDMA	Wideband code division multiple access
WiMedia	Wireless multimedia
WLAN	Wirless local area network
ZOH	Zero-order hold

1. Introduction

1.1 Motivation

After rapid growth over last decades, wireless communication has become ubiquitous in daily life. The proliferation of wireless systems is closely coupled with the phenomenal success of CMOS technology scaling that makes it possible to develop increasingly complicated systems on a single silicon chip. It can largely be attributed to the numerous efforts related to circuit and system development to leverage the technology scaling so that it delivers the required performance and functionality at an ever lower cost, lower power consumption and smaller product size. These efforts have led to abundant innovations in integrated circuit (IC) design, both digital and analog, which underpin the evolution of wireless communication systems in the market. However, an important trend in IC design, for wireless communication and others, is the continued migration from the analog domain to the digital domain in order to avoid some fundamental weaknesses of advanced CMOS technology for analog design and to fully capitalize on the prevailing benefits of technology scaling for digital circuitry [1, 2].

The radio frequency (RF) functions for wireless communication have traditionally been realized using analog or analog-intensive circuitry. The need to process high-frequency analog signals and the demanding performance requirements set a barrier that is difficult to surmount using a digital approach. However, this barrier appears to be eventually crumbling with the improvement of digital computation capability as a result of the continued downscaling of CMOS technology. Some digital or digital-intensive RF solutions have recently proved feasible. An impor-

tant and well-celebrated example is the the low-noise digital architecture of a phase-locked loop (PLL), which proved capable of meeting the stringent requirements for wireless frequency synthesis [3, 4]. In these digital PLLs, all the building blocks are either purely digital or have digital input/output signals. The term all-digital PLL (ADPLL) has been widely adopted to highlight the extent of their digital nature and differentiate them from previous analog-oriented versions. By deliberately avoiding traditional analog circuit blocks, such as the charge pump and passive loop filter, the ADPLL architectures circumvent major design issues, including charge-pump current mismatch, capacitor current leakage and low voltage headroom, as encountered when using a conventional PLL design in ultra-scaled CMOS, and they bring the benefits of high-level integration and improved programmability associated with a digital approach. The architecture, however, is far from being mature, and it is clear that its full potential can only be tapped with new circuit-level and architectural innovations.

There are several aspects where improvement is desired for the current ADPLL architecture. The first and probably most important is its relatively high in-band phase noise level as compared with well-designed traditional analog PLLs, which is the effect of time quantization in the feedback path. The high in-band noise level could potentially limit the signal-to-noise ratio (SNR) and thus data throughput in a wireless system. Second, the ADPLL architecture still entails significant levels of power consumption and silicon area in a nanoscale CMOS, which allows for further power and cost reductions through circuit innovations. Finally, the ADPLL architecture has been almost exclusively applied to narrowband wireless systems. With wideband systems expected to be more popular as wireless communication evolves, applying the ADPLL architecture to those systems is certainly worth considering. It is, therefore, the goal of this research work to find and test techniques for improving different properties of the ADPLL architecture and for exploring its design for both narrowband and wideband wireless applications.

1.2 Research contribution

During the course of research work presented in this thesis, various circuit techniques have been developed by the author. These circuit tech-

niques affect the major building blocks and also the overall ADPLL architectures, and they have been verified with extensive simulations and measurements. Particularly, three chips have been designed by the author using a 65-nm CMOS technology. The first is an ADPLL chip characterized by a relatively narrow frequency tuning range, targeted for the 2.4-GHz ISM band application. The second is an ADPLL chip with a wide tuning range, targeted for cognitive radio applications. And the third chip is another wideband ADPLL chip, designed for verifying several architectural and circuit-level innovations. The implemented ADPLLs achieved state-of-the-art performance. The techniques and implementation results have been reported by the author in three journal papers and six conference papers [5–13]. Moreover, the author has been granted two ADPLL-related patents [14, 15].

In the first ADPLL chip design [10, 12], the major techniques that were developed include a high-speed topology for the variable phase accumulator (VPA) and a simple power-saving technique for the time-to-digital converter (TDC) core. The VPA and TDC are a few of most critical building blocks for the ADPLL. The high-speed topology improved the VPA's capability of handling a high-frequency signal fed back from the ADPLL output with a pure synchronous digital design flow and at no additional cost in terms of area or power consumption. The power-saving technique was based on a short delay line introduced in the reference signal path. The simple technique allows the TDC core to operate at a low duty cycle with a dramatic (e.g. 95% in the specific implementation) reduction in its average power dissipation.

In the second chip [7, 8], the ADPLL architecture was designed for a cognitive radio sensor radio scenario, where wide frequency range is required with a relatively moderate noise performance requirement. To the best of our knowledge, it was the first published attempt at a wideband ADPLL design. A novel wideband ring oscillator was developed, and circuit techniques such as adaptive frequency calibration based on a binary search algorithm were employed to achieve the target.

Most of the circuit innovations were developed when designing the third ADPLL chip [13, 15]. At the architectural level, a separation of coarse-tune and fine-tune branches results in a word length reduction for both of them and allows the coarse tuning logic to be powered off or clock gated during normal operation, which led to a significant reduction in the area

and power consumption for the digital logic and simplified the digital design. Meanwhile, the idea of frequency reference multiplication was implemented to lower the effect of quantization noise arising from the TDC. The TDC was designed to be tunable to obviate the need for costly arithmetic multipliers in the digital calibration logic. A novel multi-path delay line was developed to allow the TDC to achieve a fine time resolution beyond the intrinsic delay of an inverter. In addition, an original technique was employed for the delay tuning of the TDC and the frequency tuning of the wideband ring oscillation, which is characterized by a compact design and excellent linearity.

1.3 Organization of the thesis

This thesis is organized as follows. In Chapter 2, the general requirements and different approaches of wireless frequency synthesis are presented. It is not intended to be a comprehensive study of frequency synthesis fundamentals but to provide sufficient background for the rest of the chapters. Chapter 3 gives the fundamentals of the ADPLL architecture and provides a review of the latest developments with respect to the subject. Chapter 4 covers frequency acquisition techniques that are applicable to ADPLLs. The design principles of two key building blocks, the digitally controlled oscillator and the time-to-digital converter (TDC), are respectively presented in Chapter 5 and Chapter 6. The prototypes and experimental results are presented in Chapter 7. Finally, Chapter 8 offers some conclusions.

2. Frequency synthesizer in wireless communication

2.1 Introduction

A key function of a radio transceiver in wireless communication is frequency translation. The baseband signal is up-converted in the transmitter to a radio frequency (RF) for radiation through the antenna, while the incoming RF signal from the antenna is down-converted in the receiver for information extraction in the baseband. The need for frequency translation stems from several reasons. First, the baseband frequency is generally too low for effective radiation and reception through the antenna of a suitable size, which needs be a significant fraction of the signal wavelength. Besides the antenna consideration, frequency translation also allows the same air space to be shared for the simultaneous propagation of multiple signals with originally overlapping frequency components, by translating them into different channels in the radio spectrum. In other words, channel selection and data modulation are often performed along with frequency translation.

A frequency synthesizer, commonly serving as the local oscillator (LO) in a radio transceiver, provides an accurate frequency reference for the frequency translation. The quality of the frequency reference to a large extent determines that of frequency-translated signal, and it has a critical impact on the overall performance of the transceiver and the entire wireless system. Consequently, the design of frequency synthesizers needs to meet stringent requirements imposed by the system-level specifications outlined in corresponding industrial standards. The performance requirements, along with the consideration of cost and power consumption, dictate the choice of frequency synthesizer architecture and design

methodology. This chapter reviews frequency synthesizer performance requirements from the system-level perspective of wireless communication.

2.2 Radio transceiver architectures

Radio transceiver architectures are often categorized based on the frequency translation scheme that has been employed. Depending on whether an intermediate frequency (IF) is involved in the frequency translation between the baseband and RF and the relative location of the IF, three well-known architectures can be identified: superheterodyne, direct conversion and Low-IF architecture.

2.2.1 Superheterodyne

In a superheterodyne receiver, the frequency down-conversion is performed in two or more steps, first from the RF to a relatively high IF and then from the IF to the baseband with or without more IFs involved. Likewise, frequency up-conversion in a super-heterodyne transmitter consists of baseband-to-IF and IF-to-RF up-conversions involving two or more steps. Fig. 2.1 shows a generic block diagram of a superheterodyne receiver with two-step down-conversion. The received RF signal is first passed through the band-selection filter to attenuate the out-of-band blockers. It is then amplified by the low-noise amplifier (LNA) to overcome the noise contributions from the subsequent stages. The image-reject filter is meant to further suppress any unwanted signals and noises in the image band, which would be superimposed on the desired signal through the RF-to-IF down-conversion. The mixing with the first LO signal from the RF synthesizer then down-converts the signal to an IF. For channel selection, the LO signal frequency needs to be tunable so that the selected channel will be centered at the fixed IF after the down-conversion, regardless of the channel location in the reception band. The IF signal then goes through the channel-selection filter, which passes the selected channel and rejects out-of-channel energy. The signal level of the IF signal is then properly adjusted by a variable gain amplifier (VGA) prior to the IF-to-baseband down-conversion, which is performed by the quadrature mixer supplied with a quadrature LO signal at the fixed frequency. The ultimate channel selection is then performed by the low-pass baseband filters in the in-phase (I) and quadrature-phase (Q) paths. The baseband VGAs then

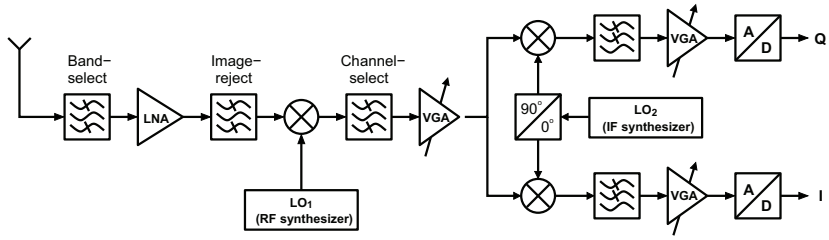


Figure 2.1. Block diagram of a superheterodyne receiver.

adjust the signal voltage swing to an optimal level for later analog-to-digital (A/D) conversion.

The superheterodyne architecture, especially the receiver, is well known for its capability to deliver excellent performance, and it was the dominant choice in wireless applications for decades. However, this architecture requires expensive discrete components, e.g. high-Q RF filters, and it does not allow for high-level integration with current IC fabrication technologies. It is also power hungry because driving the external components tends to incur significant power consumption. As a result, the interest in this architecture has drastically diminished in recent years.

2.2.2 Direct conversion

Direct conversion transceiver architecture is characterized with one-step frequency translation, either a baseband-to-RF translation or an RF-to-baseband translation. Since no IF is involved, it is also referred to as zero-IF architecture. Outperformed by the superheterodyne architecture, the architecture received little attention after its inception for several decades until it was eventually adopted for the radio-paging receiver in 1980 [16]. However, this architecture has the advantage of high integrability as well as low power consumption in comparison with the superheterodyne. With the aggressive demand for high-level integration from the market, it has received increasing interest with more real implementations [17–19].

Fig. 2.2 shows a block diagram of a direct conversion receiver. The received RF signal, after amplification by the LNA, is directly down-converted to the baseband via quadrature mixing. The quadrature down-conversion also provides image rejection. As the image is the desired channel itself, the I/Q matching required for image rejection is substantially relaxed in comparison to the superheterodyne scenario and thus achievable for many applications. This obviates the need for an image-reject filter. A well-

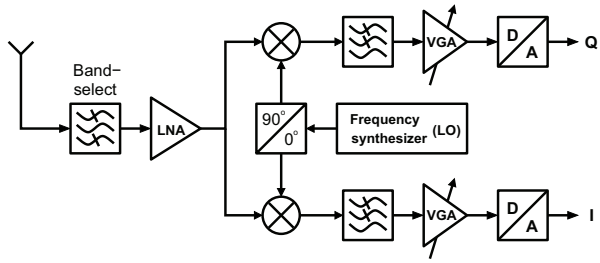


Figure 2.2. Block diagram of a direct conversion receiver.

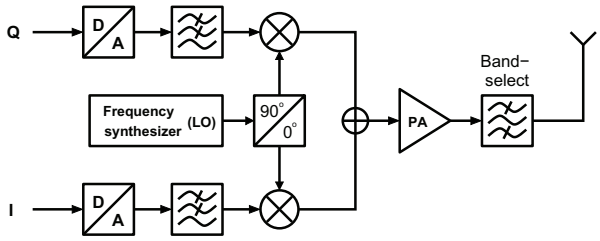


Figure 2.3. Block diagram of a direct conversion transmitter.

known issue with the direct conversion receiver is its high sensitivity to the DC offset and low-frequency flicker noise. Techniques such as AC coupling and servo loop cancellation could be employed to mitigate the issue.

Fig. 2.3 shows a block diagram of a direct conversion transmitter. Basically, it has a reverse structure of the direct conversion receiver, with A/D conversion replaced by the D/A and LNA replaced by the power amplifier (PA). The frequency synthesizer provides the LO signal at the desired carrier frequency for the baseband-to-RF up-conversion prior to the power amplification. A major issue with the direct conversion transmitter is LO pulling due to the feed-through of the PA output back to the LO, which can be mitigated or eliminated by techniques such as oscillator frequency offsetting among others [20].

2.2.3 Low-IF

Another popular architecture is low-IF architecture. This architecture also allows for high-level integrability, while it is less affected by the major issues encountered with the direct conversion architecture, such as sensitivity to low-frequency interference in the receiver and LO pulling in the transmitter. In the low-IF receiver, the desired RF channel is down-converted to a very low frequency region near the DC. The IF-to-baseband down-conversion could be performed either in the analog domain or digitally. The low-IF transmitter is often called a two-step transmitter. With

the PA output frequency shifted away from the LO output frequency, the two-step transmitter is less likely to have the LO pulling issue than the direct-conversion architecture. A major drawback of a low-IF architecture compared with the direct-conversion architecture is that matching between the I and Q branches becomes critical in order to achieve sufficient image rejection.

The above discussion in this section covers the general aspects of radio transceiver architectures. Additional information on transceiver details and their building blocks can be found in a number of published papers and books [21–23].

2.3 Requirements for frequency synthesizers

Regardless of the transceiver architectures, the fundamental role of the frequency synthesizer is LO signal generation for frequency translation. Along with this role, the requirements for the frequency synthesizer remain more or less unchanged. This section starts by discussing the basics of frequency translation, and then proceeds to give an overview of those requirements for a frequency synthesizer.

2.3.1 Frequency translation with an ideal LO signal

Frequency translation, realized by means of mixing or other similar methods, can be modeled as time-domain multiplication or equivalently as frequency-domain convolution. With $X_i(f)$ being the spectrum of the input signal and $X_{\mathrm{ref}}(f)$ the LO signal, the spectrum of the mixing result can be expressed as,

$$X_m(f) = X_i(f) \otimes X_{\mathrm{ref}}(f). \tag{2.1}$$

When only real signals are considered with a single mixer, an ideal LO signal is a pure sinusoid tone. In the frequency domain, it can be represented by a pair of Dirac impulses, i.e. $X_{LO}(f) = \delta(f \pm f_{LO})$ [1]. With this LO signal, the convolution in Eq. (2.1) can be rewritten as,

$$X_m(f) = X_i(f \pm f_{LO}). \tag{2.2}$$

[1] The amplitude has been ignored for the sake of convenience.

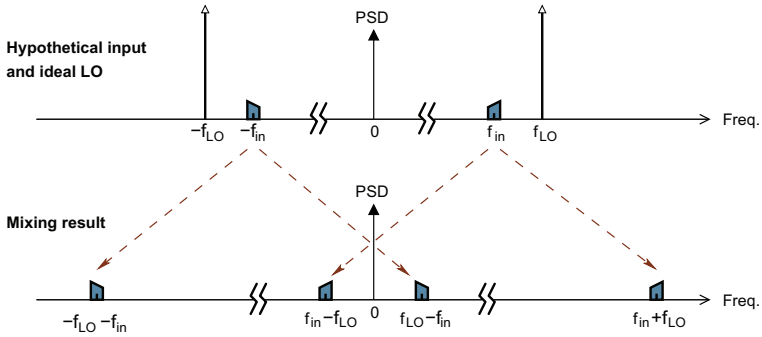


Figure 2.4. Frequency translation with an ideal real LO signal.

It shows that two replicas of the original signal, translated by $\pm f_{LO}$ respectively, are produced in the frequency domain. This is illustrated in Fig. 2.4 for a hypothetical input signal spectrum centered at frequency $\pm f_{\text{in}}$. In this hypothetical scenario, frequency up-conversion and down-conversion can both be readily achieved by accordingly filtering out the unwanted replica.

An important consideration with real signal mixing is the image spectrum phenomenal. As can be seen from Eq. (2.2), two spectra located at equal distance but on different sides of the LO frequency have an overlapping replica after mixing. This accounts for the need of image-reject filtering before the frequency translation or image-reject mixing with a quadrature down-converter and up-converter in order to prevent the image spectrum from corrupting the desired channel through frequency translation. The image-reject mixing can be considered complex signal mixing, where the LO signal with quadrature phase splitting represents a single Dirac impulse instead of two impulses in the frequency domain.

2.3.2 Frequency synthesizer performance metrics

A real world frequency synthesizer is characterized by a number of non-idealities. Some of the non-idealities are unimportant or can be readily remedied with other functional blocks usually presented in the radio transceiver. For example, while the waveform of a frequency synthesizer is usually different from a sinusoid, it is still simply a periodic function at a fundamental frequency, f_{LO}. Such a waveform has the high-order harmonics of the fundamental frequency, and its spectrum resembles an impulse comb instead of a single impulse or pair of impulses. Since the harmonics are generally considered to be far apart from each other, the as-

sociated image spectra for mixing or the resulting spectrum replicas can usually be readily filtered out, and thus they are not a concern. In fact, the LO signal is usually formed using buffers or frequency dividers into a square wave as an input to the mixer. Another example is the amplitude variation of the frequency synthesizer output, which is often neglected since it is usually negligible and can essentially be eliminated by using a limiter at the output [24].

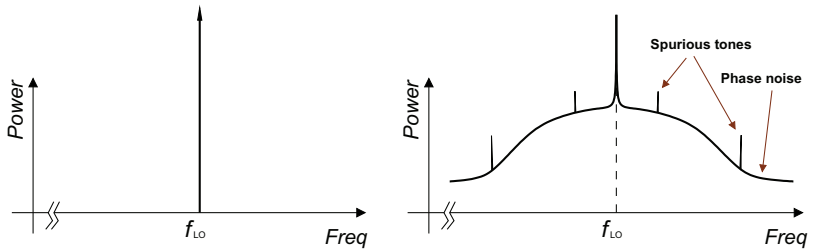
On the other hand, there are usually stringent requirements for the spectrum purity around the fundamental frequency. This spectrum purity is commonly quantified with two critical performance metrics, i.e. phase noise and spurious tones. In addition to the spectrum purity requirement, the frequency synthesizers are supposed to be programmable so that they will provide the desired output frequency accurately and promptly at any time of request for any of the target channels. The frequency tuning range, frequency step and frequency settling time are the common performance metrics used to characterize the programmability of a frequency synthesizer.

2.3.2.1 Phase noise and spurious tones

With the high-order harmonics and amplitude noise ignored for simplicity, a general frequency synthesizer output could be expressed in the following sinusoidal form in the time domain,

$$x_{LO}(t) = \mathrm{A} \cdot \cos(2\pi f_{LO} t + \phi(t)), \tag{2.3}$$

where the amplitude A is considered a constant and $\phi(t)$ represents the phase fluctuation over time. The phase fluctuation can be attributed to various non-idealities with a real frequency synthesizer implementation, such as physical noise sources, component mismatches, quantization effects and nonlinearities. It generally comprises two different types of components, random and periodic. Phase noise is defined as the random phase fluctuation, while spurious tones, also called “spurs”, refer to the periodic components in the phase fluctuation. The phase fluctuation spreads the signal power over the neighboring frequency region and creates sidebands around the desired tone in the frequency domain. Fig. 2.5 illustrates the effect of phase fluctuation, where phase noise and spurious tones can be identified respectively as the continuous noise skirt and discrete spikes above the noise.



(a) Ideal output **(b) output with phase fluctuation**

Figure 2.5. Phase fluctuation effect in frequency domain.

The phase noise is commonly specified in terms of the single sideband (SSB) power spectral density (PSD) at a given offset frequency normalized to the total carrier signal power. It can be expressed as

$$\mathcal{L}\{f\} = 10 \cdot \log \frac{PSD\{f_{LO} + f\}}{P_{total}}, \tag{2.4}$$

with units of dBc/Hz, i.e. decibels relative to the carrier per Hertz. This phase noise, $\mathcal{L}\{f\}$, is often plotted as a function of the offset frequency at a logarithmic scale to give a more complete picture of the overall phase noise performance. In addition, it may also be integrated over a specific range of offset frequencies to obtain the integrated phase noise. More specifically, the integrated phase noise is given by

$$\mathcal{L}_{int} = 2 \int_{\mathrm{f1}}^{\mathrm{f2}} \mathcal{L}(f)\, df, \tag{2.5}$$

which has units of dBc, i.e. decibels referenced to the carrier. The factor 2 in the formula accounts for the fact that the integration is performed on a single sideband. This is illustrated with a phase noise plot in Fig. 2.6. Note that the integrated phase noise also includes the power of spurious tones within the frequency range. Another commonly used measure of phase noise, closely linked to the integrated phase noise, is the root-mean-square (RMS) phase error. Its relationship with $\mathcal{L}(f)$ is given by the following equation [25],

$$\sigma_\phi = \frac{180}{\pi}\sqrt{\mathcal{L}_{int}} = \frac{180}{\pi}\sqrt{2 \int_{\mathrm{f1}}^{\mathrm{f2}} \mathcal{L}(f)\, df}, \tag{2.6}$$

in units of degree. Alternatively, the RMS phase error is expressed as jitter in time domain. The RMS jitter in units of second is related to the

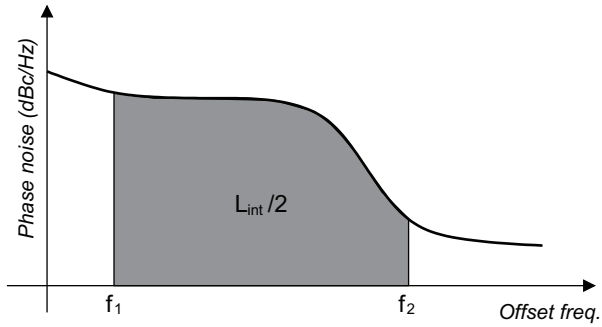


Figure 2.6. Phase noise plot and integrated phase noise.

RMS phase error in degrees as

$$\sigma_j = T_0 \cdot \sigma_\phi / 360, \tag{2.7}$$

with T_0 being the carrier period in seconds. Likewise, given the jitter information, the RMS phase error can be calculated as $\sigma_\phi = f_0 \sigma_j / 360$, with f_0 being the carrier frequency. Representing spurious tones is often more straightforward. They are usually specified as the power ratio of the tone at a given offset frequency to the carrier power and have the unit of dBc.

Phase noise and spurious tones have similar impacts on the transceiver performance. In general, they degrade the receiver's sensitivity and selectivity, and cause modulation inaccuracies and out-of-band spurious emissions in the transmitter. Fig. 2.7 shows an example of RF-to-IF down-conversion in a superheterodyne receiver with phase noise and spurious tones present in the frequency synthesizer output. The received signal spectrum consists of a weak desired channel and several stronger adjacent channels. As the LO frequency down-converts the desired channel to IF, the adjacent channels are also down-converted to IF by the corresponding LO sideband phase noise (indicated with shades area) and the spurious tones. The down-converted adjacent channels fall into the same frequency range as the desired channel. Consequently, the signal-to-noise ratio (SNR) is severely lowered through the down-conversion. This raises the bit error rate (BER) in the recovered data. If the phase noise and spurious tones are not sufficiently low, the desired signal may be totally overwhelmed by the blockers in adjacent channels and might not be detected at all, depending on the relative power of adjacent channels.

In the transmitter, the signal spectrum before the up-conversion can usu-

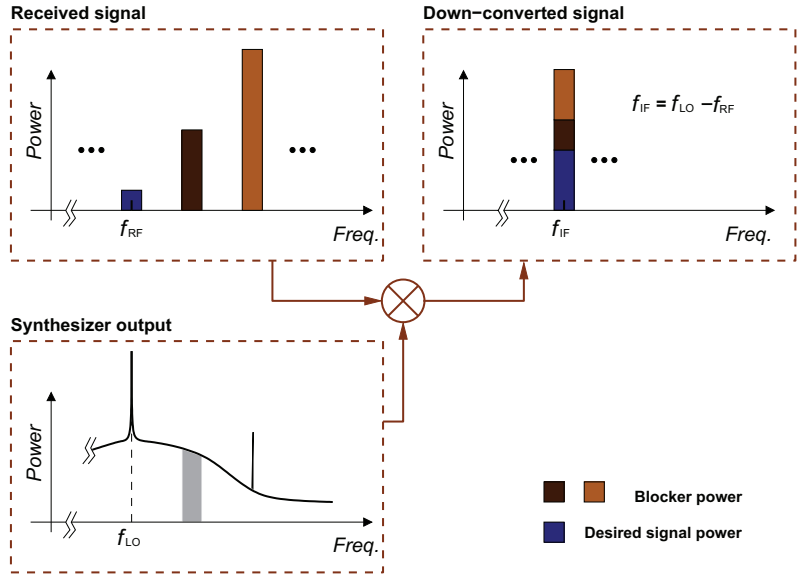


Figure 2.7. Effect of phase noise and spurious tones in a receiver.

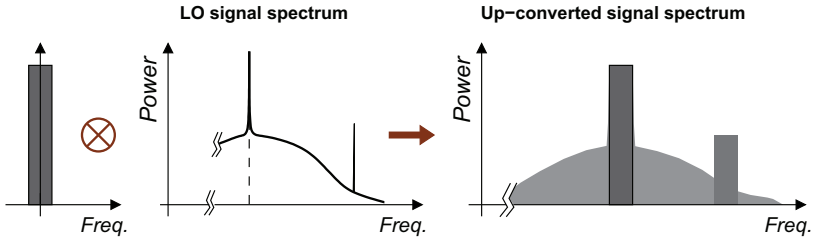


Figure 2.8. Effect of phase noise and spurious tones in a transmitter.

ally be well defined and band-limited with appropriate filtering. With an ideal LO signal, the mixing process involves merely up-shifting the spectrum in its entirety with no change to its bandwidth. However, with an LO signal from a real frequency synthesizer, the presence of phase noise and spurious tones spread the signal spectrum beyond its original bandwidth. This is illustrated in Fig. 2.8 for a direct conversion transmitter, where the baseband signal is directly up-converted to the RF. The spreading of signal power beyond the channel boundary creates interference in the adjacent channels. To limit this adjacent channel interference in a wireless system, a spectral mask is commonly specified in the corresponding wireless standard to regulate maximum power emission allowed for the transmitter over different frequencies. The phase noise and spurious tones must be sufficiently low for a transmitter to satisfy the emission spectrum mask. In general, for systems with narrow channel spacing such as GSM, the requirements for spectrum purity tend to be more stringent.

In addition to the impact on the up-converted and down-converted signal spectrum, another related but different impact of phase noise and spurious tones on radio transceiver performance comes through the phase modulation and demodulation steps, which are commonly performed along with the frequency translation, e.g. by means of quadrature mixing. This impact can be readily understood by considering the phase error in a constellation diagram. Fig 2.9 shows a constellation diagram of a quadrature phase shift keying (QPSK). Assume that the system is otherwise ideal, with the only error contributor being the phase fluctuation of the LO signal. The intended symbol in this example is "11", with its ideal position in the constellation diagram represented by the darkened dot. However, due to the non-zero phase error, ϕ, of the LO signal, the actual received signal is located at the center of the unfilled circle nearby. The displacement of the actual position from the ideal symbol position, represented by the error, E, in this example, can be computed from the phase error as,

$$E = 2r\sin(\frac{\phi}{2}), \tag{2.8}$$

where r is the radius of the large circle, which in this example equals 1. Assuming that ϕ is small, it can be approximated as $E \approx (\pi\, r\, \phi)/180$, with ϕ given in degrees. The error vector magnitude (EVM), which is defined as RMS percentages, can be approximated as [25]

$$EVM \approx 100\% \left(\frac{\pi}{180}\right) \sigma_\phi, \tag{2.9}$$

where σ_ϕ is the RMS phase error given in Eq. (2.6). From here, it is clear that the phase error, which is a measure of phase noise and spurious tones combined, distorts the constellation and thus degrades the system performance. The EVM is usually an important figure of merit specified in different wireless standards. Besides the phase error, it has various other contributors, such as IQ imbalance, in the systems. The phase noise and spurious tones from the frequency synthesizer should be sufficiently low so that the phase error from the LO signal only contributes a small fraction of the standard specified EVM.

2.3.2.2 Tuning range and frequency resolution

The tuning range of a frequency synthesizer is simply the frequency range over which the synthesizer output can be tuned. When designed for a single-band wireless system, this frequency range needs to cover all the

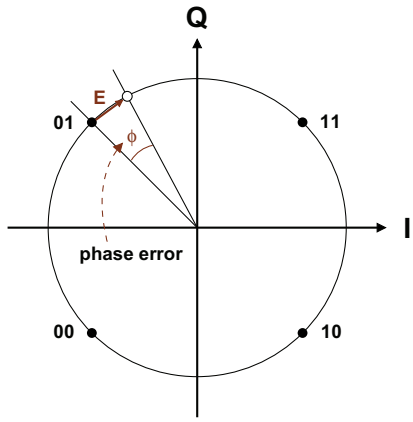


Figure 2.9. Effect of phase error on a constellation diagram.

frequency channels in the target frequency band. For a multiband system or multi-system applications, it needs to cover all of the relevant frequency bands. With the growing popularity of wideband and multiband systems, it is common that frequency synthesizers need to have a very wide frequency range, which tends to be difficult or costly to achieve with existing techniques. With a PLL frequency synthesizer, the tuning range is usually determined by its on-chip oscillator. Wide-range tunable oscillators remain an important research subject and will be treated in more detail in Chapter 5.

Frequency resolution, or frequency step size, refers to the smallest frequency increment possible when using the synthesizer tuning. The upper bound of frequency resolution is usually set by system channel spacing. The frequency resolution needs to be equal to or smaller than the channel spacing such that all of the channel center frequencies can be synthesized with sufficient accuracy.

Some examples of frequency range and channel spacing in cellular and short-range wireless systems are given in Table 2.1.

Table 2.1. Channel spacing in some wireless communication systems

	GSM	GPRS	DECT	WCDMA	Bluetooth	Zigbee
Frequency range (MHz) Rx/Tx	925-960 880-915	925-960 880-915	1880-1900	1920-1980	2402-2480	2405-2480
Channel spacing (kHz)	200	200	1728	5000	1000	577

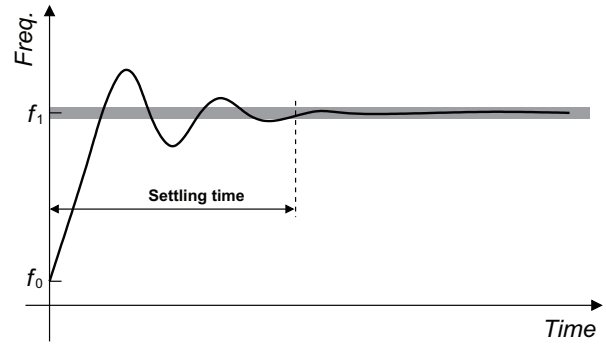


Figure 2.10. Settling time of a frequency synthesizer.

2.3.2.3 Settling time

Settling time, also called “locking time” or “switching time”, is the time it takes to switch from one frequency to another within a given tolerance. This is illustrated in Fig. 2.10, where f_0 and f_1 represent the initial frequency and the new target frequency respectively. The narrow darkened region around the target frequency, f_1, indicates the required frequency accuracy, usually specified in a corresponding wireless standard. Since this settling time may vary in its value with different initial or target frequencies, it is typically represented by its worst-case (maximum) value for all possible scenarios.

The need for a short settling time is obvious considering the role of the frequency synthesizer in wireless systems. During the settling time, data transmission and reception are usually not possible due to the LO frequency error. The time wasted leads to a reduced data rate and extra power consumption by the system. However, the specific requirements on the settling time to fulfill system specifications may vary substantially from one system to another. The most stringent requirements probably come from the systems where fast frequency hopping is employed to minimize interference and multi-path fading and to protect data security. Since the frequency synthesizer needs to be switched for every short amount of time, the time window for settling is quite constrained. An extreme example is the WiMedia ultra wideband (UWB) system, where rapid frequency hopping at a 3-MHz hopping rate demands a settling time of less than 10 ns. This extremely short settling time is still considered unfeasible with a simple PLL frequency synthesizer and entails some unique techniques and additional devices [26]. Another example is the GSM system, where a relatively moderate settling time of less than

150 μs (0.1-ppm frequency accuracy) is required. In the 2.4-GHz ISM band WLAN system, the required settling time is even more relaxed, i.e. 224 μs with a 20-ppm frequency accuracy.

2.4 Frequency synthesis approaches

There exist different approaches for frequency synthesizer realization, which can generally be classified as direct analog frequency synthesis, direct digital frequency synthesis, delay-locked loop (DLL) based frequency synthesis and PLL-based frequency synthesis. General discussions on these frequency synthesis techniques can be found in the literature [27–31]. Among these approaches, PLL-based frequency synthesis is the most universal and dominant choice in wireless communication systems thanks to the relative strength and flexibility of PLL frequency synthesizers.

2.4.1 Direct analog synthesis

Direct analog frequency synthesis (DAS) generates a desired frequency from a single fixed reference frequency by employing frequency multiplication, mixing and division [28]. This approach is illustrated in Fig. 2.11. The multiple fixed reference frequencies at the input can be generated using division and multiplication with a single precision frequency reference. For example, a 5-MHz crystal oscillator can be employed with its output first divided down to 1 MHz and then multiplied to 3 MHz and $27 \cdots 36$ MHz. Depending on the control word variable value, a_0, the first switch accordingly selects one of the frequencies from 27 to 36 MHz and passes it to the first mixer followed by a band-pass filter. The result is that the selected frequency is added to 3 MHz. The subsequent divider then shifts down the frequency by a factor of 10, resulting in a frequency with a corresponding refined resolution. As illustrated in the figure, the process can be repeated a given number of times to produce a tunable output frequency with a sufficiently fine resolution. The output frequency in this example can be expressed as

$$f_{out} = 3 + a_n + a_{n-1} \cdot 10^{-1} + a_{n-2} \cdot 10^{-2} + \cdots + a_1 \cdot 10^{-(n-1)} + a_0 \cdot 10^{-n} \text{ (MHz)}. \tag{2.10}$$

One advantage of DAS is that it can switch the output frequency rapidly.

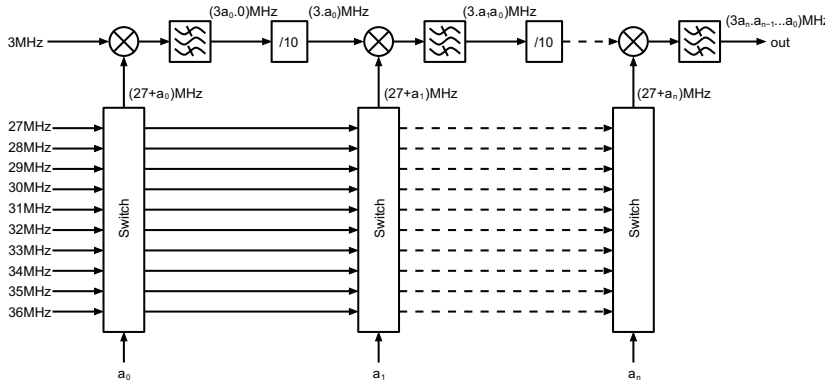


Figure 2.11. An example of direct analog frequency synthesis.

With adequate electrical isolation between the stages, excellent output spectrum purity is achievable [31]. However, the DAS is usually limited to a relatively low output frequency and coarse frequency resolution as a result of its mix-and-divide approach, which is evident in the above example. Moreover, frequency synthesizers of this type tend to be excessively bulky and power hungry, making them unfit for on-chip integration. These drawbacks of DAS make it almost non-existent in wireless applications.

2.4.2 Direct digital synthesis

The basic principle of direct digital frequency synthesis (DDS) is illustrated in Fig. 2.12. The frequency control word (FCW) is fed to the phase accumulator, defining its increment value for every clock (clk) cycle. The phase accumulator output represents the instantaneous phase of a sinusoid waveform. It is then converted into the corresponding amplitude by the subsequent phase-to-amplitude converter (PAC), which typically implemented as a look-up table in read-only memory (ROM). The phase accumulator and phase-to-amplitude converter constitute a numerically controlled oscillator (NCO), whose output is a sine wave in the digital domain. The digital-to-analog converter (DAC) transforms the digital waveform into an analog signal, which is then passed through the low-pass filter (LPF) to smooth the waveform and remove its spurious tones and harmonics, mostly those arising from the D/A conversion. The output frequency of DDS can be expressed in terms of the clock frequency as

$$f_{out} = \frac{\Delta\phi}{2^W} \cdot f_{clk}, \tag{2.11}$$

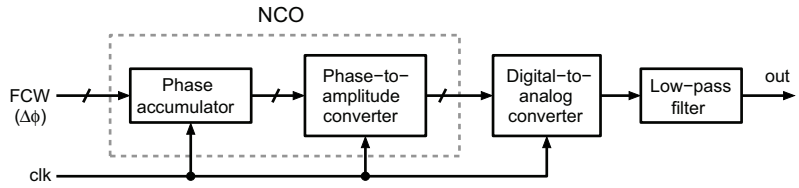


Figure 2.12. An example of direct digital frequency synthesis.

where $\Delta\phi$ is the value of the digital control word, FCW, and W the accumulator word length.

The DDS also allows for fast frequency switching, which is a common feature of direct frequency synthesis. Due to its digital-intensive nature, it lends itself well to full integration in a CMOS technology, and readily incorporates the implementation of different data modulation schemes. In principle, it can be implemented to achieve an arbitrary fine frequency resolution. However, according to sampling theory, a DDS can only synthesize frequencies at a maximum of up to half the clock frequency, that is, $\Delta\phi \leq 2^{W-1}$. In other words, the majority of the circuitry has to be clocked at a frequency at least twice that of its output, which not only severely limits its output frequency but also incurs high power consumption, even for a moderate output frequency. Meanwhile, DDS output usually has high spurious content due to the quantization and nonlinearity associated with the DAC, which constitutes a performance bottleneck [32, 33]. All of these fundamental drawbacks have essentially prohibited DDS from common use in RF systems, especially for mobile terminals.

2.4.3 PLL-based synthesis

The PLL-based approach has been the predominant choice for RF synthesis nowadays. Basically, a PLL is a negative feedback loop that in a way locks the phase of an in-loop oscillator to that of an off-loop precision oscillator. As a result, it combines the desired features of the two oscillators and overcomes some fundamental limitations with a practical stand-alone oscillator. Its output signal, tapped from the in-loop oscillator, retains the features of tunability and the high frequency of the latter, while at the same time acquiring long-term frequency precision and stability from the off-chip oscillator. Before the advent of ADPLLs, PLL frequency synthesizers typically employed charge-pump architectures, which are usually grouped into two types, integer-N and fractional-N PLLs, depending on the possible ratios of the output frequency to the reference frequency. Es-

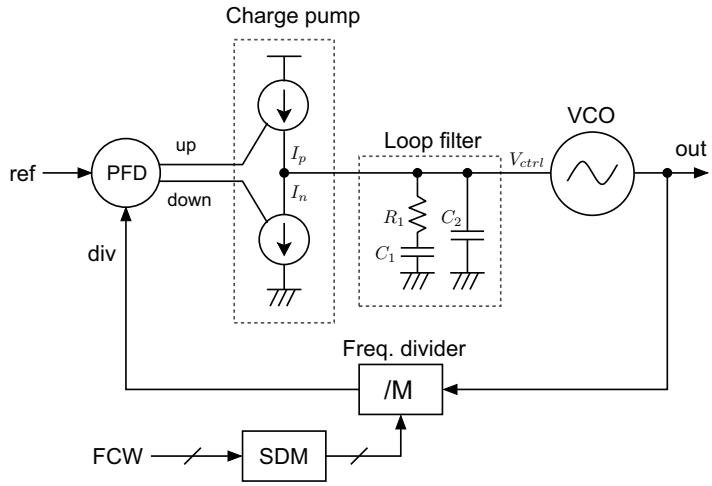


Figure 2.13. An example of a charge-pump PLL.

sentially, the latter can be viewed as a generalization of the former or the former can be viewed as a simplification of the latter.

In most wireless applications, an integer-N frequency synthesizer usually cannot satisfy the overall performance requirement. Therefore, the most common PLLs for frequency synthesis belong to the fractional-N category. Fig. 2.13 shows a generic block diagram of a representative traditional fractional-N PLL [34–39]. The RF output, out, is generated by the voltage-controlled oscillator (VCO). The frequency reference, ref, is typically generated by a crystal oscillator off chip. The output frequency is defined in terms of its ratio to the reference frequency by the FCW. In the feedback path, the VCO output is passed through the multi-modulus frequency divider to generate the frequency-divided signal, div. The use of a $\Sigma\Delta$-modulator (SDM) makes it possible for the FCW to define an effective fractional division ratio. The phase difference between the frequency reference, ref, and the frequency-divided VCO signal, div, is then estimated by the phase/frequency detector (PFD) in terms of the time difference between their respective closest rising edges. The PFD generates either an "up" or a "down" pulse with a width proportional to the measured time difference. This signal from the PFD is then converted by the charge pump into a current pulse, I_p or I_n, with a proportional duty cycle. Accordingly, the VCO tuning voltage, V_t, and its output frequency increases or decreases. As a result, the VCO steady-state output frequency is equal

to the reference frequency multiplied by the division ratio, i.e.,

$$f_{out} = F_{\mathrm{div}} \cdot f_{\mathrm{ref}} = (F_{\mathrm{div,I}} + F_{\mathrm{div,F}}) \cdot f_{\mathrm{ref}}, \tag{2.12}$$

where F_{div} is the division ratio comprising an integer part, $F_{\mathrm{div,I}}$, and a fractional part, $F_{\mathrm{div,F}}$. The loop filter is needed to suppress the noise and any glitches present in the current pulses from the charge pump in order to prevent them from generating excessive noise and spurious tones in the VCO output spectrum [40].

2.4.4 DLL-based synthesis

Recently, there have been significant efforts to explore delay-locked loops (DLLs) for frequency synthesis [41–46]. The interest in DLL-based frequency synthesis can largely be attributed to the possibility of delivering superior close-in phase noise performance due to the reduced phase noise accumulation in a DLL compared with a PLL. However, DLL-based frequency synthesis still suffers from relatively high output spurs that limit its practical applications.

An early architecture of DLL-based frequency synthesizers is shown in Fig. 2.14. Each rising edge of the reference signal, ref, drives the voltage-controlled delay line (VCDL) to generate a burst of its delayed versions, which are evenly spaced over one period of the reference signal when the DLL is in locked state. These rising edges are then combined to form a pattern of higher frequency transitions of the desired RF output signal. Since the generation of rising edges for the output is renewed for each reference period, accumulation of edge uncertainties or phase noise through VCDL is confined to only one period of the reference signal, amounting to super close-in phase noise performance. However, static phase offsets and mismatches between the delay stages of the VCDL tend to generate spurs in the output spectrum [47]. Another problem with this edge-combining scheme is the difficulty in programming the output frequency, which makes it of little use for common wireless systems.

The frequency programmability can be achieved by replacing the edge combination with an edge selection scheme. The resulting architecture, with certain techniques demonstrated in one of our recent designs [46], allows for frequency tuning with a fine fractional resolution. With the frequency control external to the feedback loop, it also features fast fre-

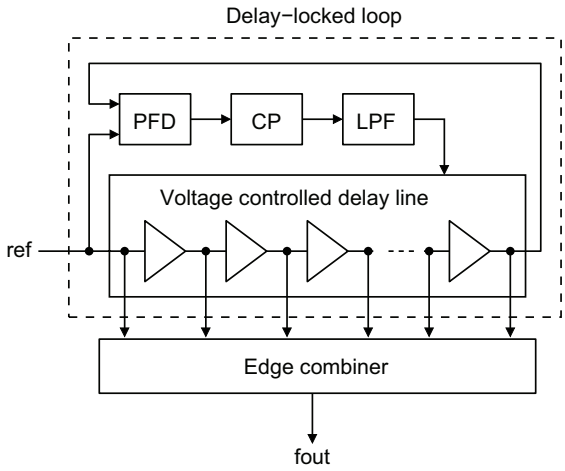


Figure 2.14. A basic DLL-based frequency synthesizer.

quency switching with a settling time of the order of nanoseconds. However, one fundamental limitation of the edge-selection DLL-base frequency synthesis is the time quantization associated with the discretely distributed rising edges of the delay line, from which the output is derived. This time quantization leads to large cycle-to-cycle jitters in the time domain when a fractional frequency resolution needs to be achieved, which translates into dominant far-off phase noise and spurs in the frequency spectrum. In addition, the feasible output frequency with a CMOS implementation of the edge-selection scheme is still limited to a relatively low part of the RF spectrum due to the speed limitation of the digital logic for the edge selection.

Alternatively, some recirculating DLL architectures have been explored for frequency synthesis [45, 48]. These types of implementations are very similar to PLLs, with an oscillator-like delay line. However, they retain the advantage of limited phase noise accumulation by realigning the phase for each reference cycle. In addition, the effect of delay mismatches is minimized since the output signal is typically derived from only one place in the delay line. However, the phase realignment errors, which arise during the process of injecting the reference edges in to the delay line, constitute a major source of output spurs. Meanwhile, it remains difficult to achieve a fine fractional frequency resolution with these kinds of implementations.

2.5 Summary

A frequency synthesizer plays an important role in a wireless system by providing a reference frequency for the frequency translation process in the radio transceiver. It usually needs to meet a set of stringent performance requirements depending on the specific application system. There are different approaches for frequency synthesis, which can be roughly divided into two categories: direct frequency synthesis and indirect frequency synthesis. The indirect approach, namely PLL-based frequency synthesis, is the predominant choice in wireless applications since it has the capacity for excellent performance, relative simplicity and low cost. Meanwhile, frequency synthesizers based on ADPLLs have become increasingly popular over last years, because they are easier to program and more compatible with the downscaled CMOS technologies. The fundamentals of ADPLLs will be discussed in next chapter.

3. All-digital PLL fundamentals

3.1 Introduction

While the PLL-based approach is the predominant choice for RF synthesis, the design of traditional PLLs is facing difficulties in using the ever downscaling CMOS. The analog building blocks, particularly the charge pump, the analog loop filter and the voltage controlled oscillator, see the nowadays nanoscale CMOS as an increasingly hostile environment for their implementation. The recent emergence of ADPLLs for frequency synthesis provides a promising path to work around this compatibility issue. An ADPLL employs digital or digital-like circuits to achieve better compatibility with the CMOS technology.

This section presents an overview of ADPLL frequency synthesizers with emphasis on a representative architecture that was adopted as the base architecture throughout the author's design work. It starts with general considerations. The frequency transfer function and phase transfer function are derived using models of discrete-time and continuous-time domains as well as their transforms. Performance analysis with respect to system requirements are provided and alternative architecture variants are then presented and compared. The forthcoming analysis has partially been presented in the literature by Staszewski and others [4, 49], but the overall analysis is the author's own work.

3.2 A generic architecture

Fig. 3.1 shows a generic ADPLL architecture, which was first reported by Staszewski et al in a previous study [4]. The RF output is generated by a DCO. The feedback path is essentially a phase-to-digital converter (PDC), which digitizes the variable phase of the DCO output with respect to that of the frequency reference and allows the subsequent loop circuitry to be pure digital logic. After digitization of the phase information, the instantaneous ADPLL output frequency is estimated with respect to the reference frequency by taking the derivative of the phase signal in the digital domain. The frequency error is then calculated by subtracting the estimated frequency from the desired frequency represented by the frequency control word (FCW), which is in turn accumulated to form the phase error. In the forward path, the phase error is conditioned with a digital loop filter (DLF) to tune the DCO frequency and correct the frequency error. As a result of the feedback mechanism, the output frequency from the DCO settles to the target frequency with a certain accuracy and noise level.

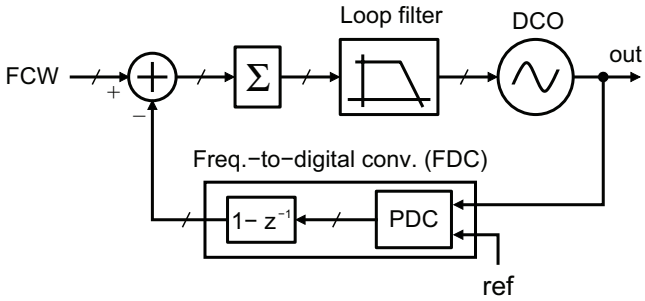


Figure 3.1. Phase-domain ADPLL architecture using frequency comparison.

3.2.1 Discrete-time operation

A digital circuit can be described in the discrete-time domain using a difference equation. Its transfer function in the z-domain can be conveniently obtained by means of z-transform. As a digital-intensive circuit, the essential operation dynamic of the ADPLL can be evaluated and analyzed in the discrete-time domain and its z-transform domain. Next, we examine the difference equations and z-domain transfer functions for different circuit blocks of the ADPLL, and then derive the overall transfer function for the ADPLL.

3.2.1.1 Digitally controlled oscillator

The DCO may be considered a digital-to-frequency converter as far as its transfer function is concerned. Neglecting noises and nonlinearities, its kth sampled output frequency can be written as

$$f_{\mathrm{dco}}[k] = K_{dco} d[k-1], \tag{3.1}$$

where K_{dco} is the DCO digital-to-frequency gain. The corresponding z-transform of the above difference equation is

$$f_{\mathrm{dco}}(z) = K_{dco} d(z) z^{-1}. \tag{3.2}$$

The unit delay between the input and output in the above equation arises from the use of registers at the DCO input to synchronize different individual bits of each control word. It does not have a counterpart in a conventional PLL, where frequency control is performed through an analog voltage without any need for synchronization. It is one of the disadvantages associated with digital frequency control. The unit delay introduces an additional phase shift to the open-loop transfer function in the frequency domain, leading to a reduced phase margin and degraded loop stability [50]. It also complicates the analysis by increasing the order of the loop transfer function. It is noted that this delay has often been conveniently omitted in the ADPLL modeling, as in a study by Staszewski and Balsara [49], which leads to inaccuracy of the corresponding model.

The phase of the DCO output is a continuous-time signal, which can be expressed as the integral of the frequency such that

$$\theta_{\mathrm{dco}}(t) = \int_0^t \omega_{\mathrm{dco}}(t) dt = 2\pi \int_0^t f_{\mathrm{dco}}(t) dt. \tag{3.3}$$

Its kth sampled value is its value at time $t = kT_R$, which can be expressed in terms of its previous sample value and the DCO frequency as

$$\theta_{\mathrm{dco}}[k] = \theta_{\mathrm{dco}}[k-1] + 2\pi \int_{(k-1)T_R}^{kT_R} f_{\mathrm{dco}}(t) dt. \tag{3.4}$$

The sampling register at the input effectively performs a zero-order hold operation on the control word. It limits the DCO frequency update only at the sampling moment and keeps it essentially constant over every time interval between two consecutive sampling moments, as illustrated in Fig. 3.2. The finite integral on the right-hand side of Eq. (3.4) is the area

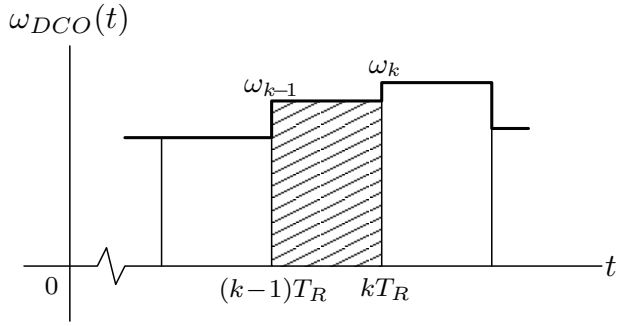


Figure 3.2. DCO phase as integral of frequency

of the shadowed rectangular in Fig. 3.2, which is $2\pi T_R f_{\mathrm{dco}}[k-1]$. This allows us to reduce Eq. (3.4) to the following,

$$\theta_{\mathrm{dco}}[k] = \theta_{\mathrm{dco}}[k-1] + 2\pi T_R f_{\mathrm{dco}}[k-1]. \tag{3.5}$$

or

$$\theta_{\mathrm{dco}}[k] = \theta_{\mathrm{dco}}[k-1] + 2\pi \frac{f_{\mathrm{dco}}[k-1]}{f_R}. \tag{3.6}$$

with T_R and f_R being respectively the reference period and frequency. The corresponding z-transform is

$$\theta_{\mathrm{dco}}(z) = \left(\frac{2\pi}{f_R}\right)\left(\frac{z^{-1}}{1-z^{-1}}\right) f_{\mathrm{dco}}(z). \tag{3.7}$$

3.2.1.2 Frequency-to-digital converter

The ideal function of the frequency-to-digital converter, excluding the quantization errors, is illustrated in Fig. 3.3. At every sampling moment with an interval of one reference period, T_R, the PDC in the feedback converts the instantaneous phase of the DCO RF output into a digital value equal to the phase normalized by 2π. The difference of phase values between the two consecutive cycles is then taken as the digital representation of the instantaneous frequency at the DCO output.

The PDC function can be represented by a difference equation and its z-transform as follows,

$$\Phi_v[k] = \frac{1}{2\pi} \cdot \theta_{\mathrm{dco}}[k] \quad \Leftrightarrow \quad \Phi_v(z) = \frac{1}{2\pi} \cdot \theta_{\mathrm{dco}}(z). \tag{3.8}$$

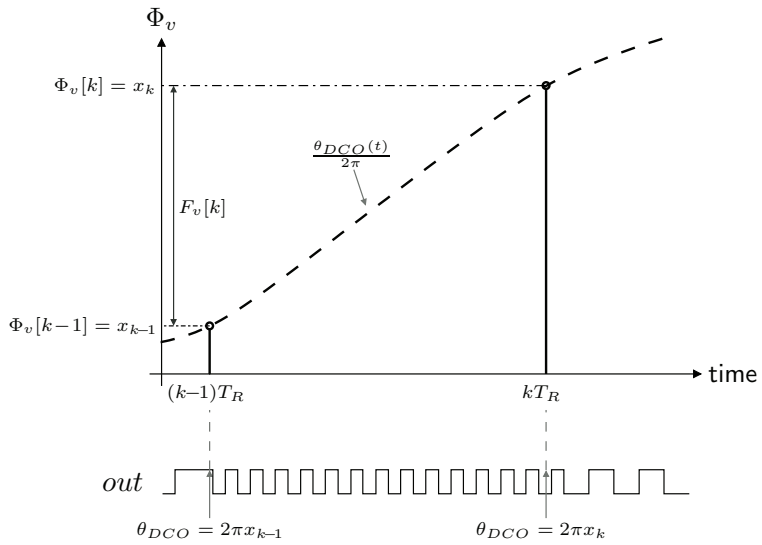


Figure 3.3. Illustration of an ideal frequency-to-digital conversion.

The subsequent difference operation is simply

$$F_v[k] = \Phi_v[k] - \Phi_v[k-1] \quad \Leftrightarrow \quad F_v(z) = \left(1 - z^{-1}\right) \Phi_v(z). \tag{3.9}$$

Combining the above result, we may write the overall difference equation for the frequency-to-digital converter (FDC) as

$$F_v[k] = \frac{1}{2\pi} \left(\theta_{\text{dco}}[k] - \theta_{\text{dco}}[k-1]\right) = \frac{f_{\text{dco}}[k-1]}{f_R}, \tag{3.10}$$

where the relationship of Eq. (3.6) has been applied. Its z-transform gives

$$F_v(z) = \left(\frac{1}{f_R}\right) z^{-1} f_{\text{dco}}(z). \tag{3.11}$$

3.2.1.3 Frequency error detector and accumulator

The frequency error detector (FED) is an arithmetic subtractor, one whose output is simply

$$F_e[k] = F_{\text{tar}}[k] - F_v[k] \quad \Leftrightarrow \quad F_e(z) = F_{\text{tar}}(z) - F_v(z), \tag{3.12}$$

where F_{tar} represents the value of the target frequency defined by the *FCW* with respect to the reference frequency. The output, $F_e[k]$, represents the frequency error in units of cycle per reference period.

Since the integral of frequency is the signal phase, the subsequent accu-

mulation, which is a digital version of the integral, generates the phase error information in the digital domain. The difference equation of the frequency error accumulator (FEA) and its z-transform can be represented as

$$\Phi_e[k] = \Phi_e[k-1] + F_e[k] \quad \Leftrightarrow \quad \Phi_e(z) = \left(\frac{1}{1-z^{-1}}\right) F_e(z). \tag{3.13}$$

It should be noted that digital computation in the FEA and the subsequent digital loop filter (DLF) usually employ saturation arithmetic so that a potentially out-of-range result would not overflow but be clamped to its nearest extreme value in the output range. Evidently, care needs to be exercised in the design to prevent saturation from occurring under normal operating conditions, where the above equations should hold true for the proper operation of the ADPLL.

3.2.1.4 Digital loop filter

The digital loop filter (DLF) is generally a low-pass filter that prevents high-frequency components of the noises passed out of the frequency error accumulator from propagating to the DCO input. Depending on the performance specifications and the level of high-frequency noise components, it can have different configurations and parameters. Its exact difference equation is determined by its specific configuration. Here, we look at some simple examples, which can serve as a basis for the study of more complicated cases.

- For a type-I PLL, there is by definition no additional integrator or accumulator in the loop other than the oscillator [51]. In the simplest case of a type-I ADPLL, the loop filter is reduced to a multiplication constant. Denote this constant as α, the difference equation is simply

$$D[k] = \alpha\Phi_e[k] \quad \Leftrightarrow \quad D(z) = \alpha\,\Phi_e(z). \tag{3.14}$$

- For a type-II PLL, there is by definition one additional integrator or accumulator in the loop other than the oscillator [51]. The simplest filter for a type-II ADPLL is a proportional-integral (PI) filter, where a digital integrator (accumulator), the integral branch, is placed in parallel with a multiplication constant, the proportional branch. The difference equations for the proportional and integral branches are, respectively,

$$D_P[k] = \alpha\Phi_e[k], \text{ and } D_I[k] = D_I[k-1] + \rho\,\Phi_e[k]. \tag{3.15}$$

The z-transforms of the proportional and integral parts are, respectively,

$$D_P(z) = \alpha\,\Phi_e(z), \text{ and } D_I(z) = \left(\frac{\rho}{1-z^{-1}}\right)\Phi_e(z).$$

The α and ρ are the scaling factors in the two branches. The overall filter output is a combination of the two:

$$D[k] = D_I[k] + D_P[k]. \tag{3.16}$$

The corresponding z-transform of the PI loop filter is

$$D(z) = D_I(z) + D_P(z) = \left(\alpha + \frac{\rho}{1-z^{-1}}\right)\Phi_e(z). \tag{3.17}$$

3.2.2 Loop transfer functions

A transfer function, which describes a transform-domain relationship between the input and output of a circuit, is a popular and powerful tool for linear analysis of a PLL. Next, we derive the overall transfer functions of the ADPLL, first for the output frequency without noise consideration and then for the phase with noise consideration included. Relevant analysis is also carried out to investigate its fundamental properties.

3.2.2.1 Frequency transfer function

Based on the above z-transform for the loop elements, we can build the overall z-transform model for the ADPLL. The result is shown in Fig. 3.4, from which we can write the transfer function for the forward path as

$$A(z) = \frac{f_{\text{dco}}(z)}{F_e(z)} = \frac{K_{\text{dco}}H(z)z^{-1}}{1-z^{-1}}, \tag{3.18}$$

and for the feedback path as

$$B(z) = \frac{F_v(z)}{f_{\text{dco}}(z)} = \frac{1}{f_R}z^{-1}. \tag{3.19}$$

The open-loop transfer function is

$$A(z)B(z) = \frac{K_{\text{dco}}H(z)z^{-2}}{f_R\,(1-z^{-1})}. \tag{3.20}$$

An important observation here is that the open-loop transfer function has at least two poles at $z = 0$, which is the result of two unit delays in the

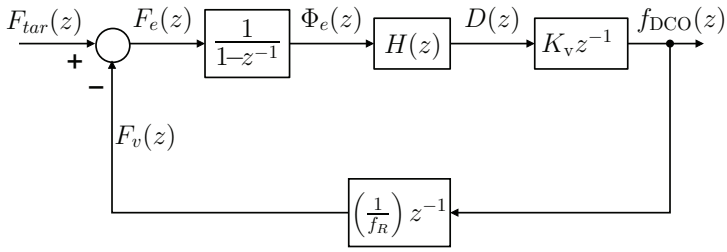


Figure 3.4. A basic z-domain model of the ADPLL.

loop. The two delays are associated with the sampling operation at the DCO input and in the PDC, which are indispensable for the ADPLL operation and thus cannot be further reduced. It should be noted that a minimum loop delay of one sampling interval, as assumed for a DPLL treated in the existing literature [50, 51], is not a realistic scenario. The ADPLL model developed in a previous study [49] has also omitted one unit delay, and is therefore not accurate. As delays in the loop adversely affect the ADPLL performance, it is in general imperative to avoid additional unit delays from being introduced into the loop filter, represented by $H(z)$ above, or any other part of the digital loop. In other words, the whole data path from the PDC output to the DLF output should not be split by any unit delay cells into more than one clock cycle. It requires that all of the intervening combinatorial logic operate fast enough to fit into one reference cycle. This might, on one the hand, demand the use of a relatively fine process node for implementation and, on the other hand, require the optimization of the digital loop circuitry to allow for timing closure.

The overall transfer function can be written as

$$G_f(z) = \frac{f_{\text{dco}}(z)}{F_{\text{tar}}(z)} = \frac{A(z)}{1 + A(z)B(z)}. \tag{3.21}$$

Apparently, the unit delays in the loop complicate the closed-loop transfer function, making it two orders higher than that of the loop filter. Substituting Eq. (3.18) and Eq. (3.19) into Eq. (3.21) yields

$$G_f(z) = \frac{K_{\text{dco}} H(z) z^{-1}}{1 - z^{-1} + K_{\text{dco}} H(z) z^{-2}/f_R}. \tag{3.22}$$

If we substitute $z = 1$ into the above equation for the DC condition, we obtain $G_f(z)|_{z=1} = f_R$, which states that $f_{\text{dco}} = F_{tar} f_R$ in steady-state operation if the loop is stable.

In general, we can assume that the DCO gain is normalized such that $K_{\mathrm{dco}} = f_R$, and accommodate the rest of the loop gain in the loop filter transfer function, $H(z)$. This allows us to rewrite Eq. (3.22) as

$$G_f(z) = \frac{f_R H(z) z^{-1}}{1 - z^{-1} + H(z) z^{-2}}. \tag{3.23}$$

As expected, the transfer function, $G_f(z)$, is proportional to the reference frequency, f_R. We may also write it as $G_f(z) = G_{f,N}(z) f_R$, where $G_{f,N}$, given by

$$G_{f,N}(z) = \frac{H(z) z^{-1}}{1 - z^{-1} + H(z) z^{-2}}, \tag{3.24}$$

is independent of f_R.

For a specific realization, we need to substitute the corresponding transfer function for $H(z)$ in the above equation. Here, we consider the two scenarios with the simplest loop filter configurations:

- In the case of using a simple multiplication constant in place of the loop filter, we have

$$G_f(z) = \frac{f_R\, \alpha z^{-1}}{1 - z^{-1} + \alpha z^{-2}}. \tag{3.25}$$

- When the PI loop filter is used, the overall transfer function becomes

$$\begin{aligned} G_f(z) &= \frac{f_R(\alpha\,(1 - z^{-1}) + \rho)\, z^{-1}}{(1 - z^{-1})^2 + (\alpha(1 - z^{-1}) + \rho)\, z^{-2}} \\ &= \frac{f_R\,((\alpha + \rho) z^{-1} - \alpha z^{-2})}{1 - 2z^{-1} + (1 + \alpha + \rho) z^{-2} - \alpha z^{-3}} \end{aligned} \tag{3.26}$$

The above examples demonstrate how cumbersome the transfer function can become if a more sophisticated loop filter is adopted.

3.2.2.2 Poles and zeros

It is well known that the transient behaviour of a linear system is completely determined by the poles and zeros of its transfer function. Therefore, it is important to consider the nature of the poles and zeros for the ADPLL in order to gain vital insight into its system behaviour.

3.2.2.2.1 Analytical solutions The example of a transfer function demonstrated in Eq. (3.25) has a pair of poles at

$$z = \frac{1}{2}\left(1 \pm \sqrt{1 - 4\alpha}\right). \tag{3.27}$$

The two poles are real and separate if $0 < \alpha < 0.25$, coincident at $z = 0.5$ if $\alpha = 0.25$, and complex if $\alpha > 0.25$. A similar discussion for this simple transfer function can also be found in the existing literature [51].

The transfer function of Eq. (3.26) has a zero at $z = 1 + \rho/\alpha$. The poles are at the roots of the denominator, which is a cubic function of the variable, z. The general analytical solutions can still be obtained, but they are not informative due to the complexity of the formulation.

3.2.2.2.2 Root-locus plots A root-locus plot, which shows the pole trajectory of the transfer function with a varying parameter, is useful for evaluating how the behaviour of the ADPLL changes with different parameters. Especially when the analytical solutions are either difficult to obtain or not informative due to their excessive complexity, root-locus plots constructed using numeric methods become more essential for facilitating the way an ADPLL is designed and characterized. Next, we examine two simple and yet representative cases.

- For a type-I ADPLL with a loop filter as a simple gain, the poles are given by Eq. (3.27). As shown in Fig. 3.5, the two poles are real when $\alpha \leq 0.25$, move towards each other as α increases, and coincide at $z = 0.5$ when $\alpha = 0.25$. They become complex conjugates when $\alpha > 0.25$ and move away from each other along a vertical line at $\Re(z) = 0.5$, which intersects the unit circle at $z = 0.5(1 \pm \sqrt{3})$ for $\alpha = 1$.

- A root-locus family is shown in Fig 3.6 for a type-II ADPLL with a simple PI loop filter. The pole originating at $z = 0$ migrates to the right on the real axis toward the zero at $z = 1$. The other two poles originate in the immediate vicinity of $z = 1 \pm j\sqrt{\rho}$, respectively.

  If $\rho = 1/4$, then the complex poles are located at the unit circle for $\alpha = 0.5$ and remain outside of the unit circle for any other α value. If $\rho > 1/4$, then the complex poles lie outside of the unit circle irrespective of α. If $0 < \rho < 1/4$, then the locus of each complex pole intersects twice with the unit circle. If $\rho = 1/27$, then three poles are coincident at $z = 2/3$ for $\alpha = 1/3$, while two of them are complex conjugates for any other value of α. If $0 < \rho < 1/27$, then the complex poles return to the real axis for some range of α, while they become complex conjugates for the rest of the α values.

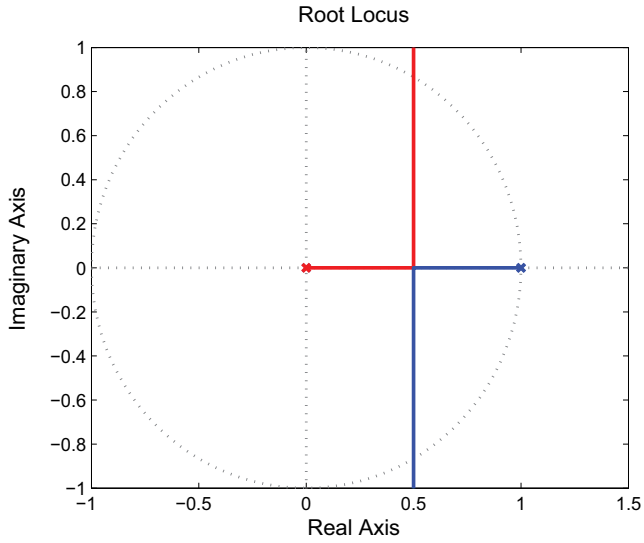


Figure 3.5. Root-locus plot of a simple type-I ADPLL.

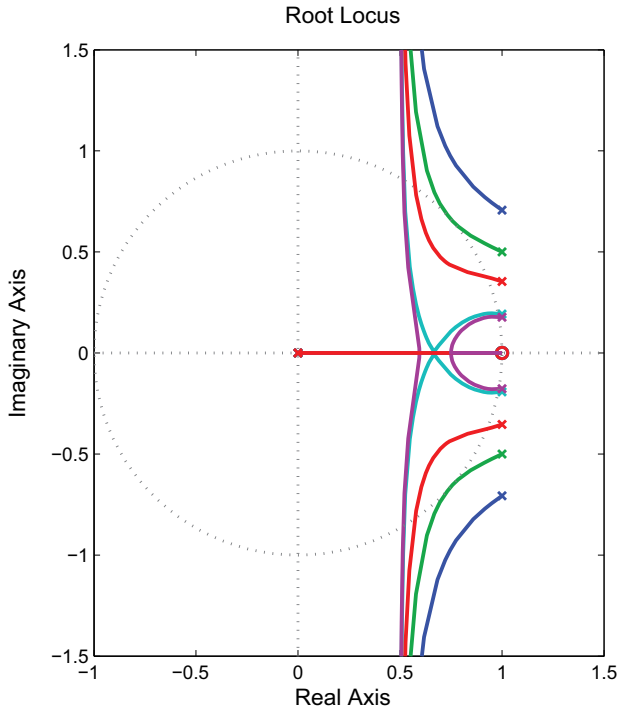


Figure 3.6. Root-locus plots of an ADPLL with a PI filter.

3.2.2.3 Loop stability

The stability requirement dictates that all of the poles of the ADPLL transfer function, Eq. (3.22) or Eq. (3.23), must lie inside the unit cycle. In this section, the stability conditions for the simplest examples of an ADPLL are examined. The effect of more sophisticated loop filter structures on the stability is also briefly reviewed.

- In the case of a first-order type-I ADPLL with a loop filter reduced to a simple gain, we can solve Eq. (3.27) for the boundary stability condition, $|z| = 1$, which gives $\alpha = 1$ and the resulting poles are located at $z = (1 \pm \sqrt{3})/2$. Therefore, the stability requirement in this case is $\alpha < 1$.

- In the case of a type-II ADPLL with a simple PI filter, the boundary condition in terms of the parameters can be obtained as follows. Considering that the stability boundary condition is characterized by the intercept of a root locus with the unit cycle, we start by defining $z = \exp(j\psi)$, with ψ being a real variable. The pole is a root of the characteristic polynomial, i.e. the denominator of Eq. (3.26). Substituting $z = \exp(j\psi)$, we obtain

$$(1 - e^{-j\psi})^2 + (\alpha(1 - e^{-j\psi}) + \rho)e^{-j2\psi} = 0, \tag{3.28}$$

  which can be rearranged to the following via some simple mathematical manipulations,

$$\alpha = \frac{4\sin^2(\psi/2)}{e^{-j\psi}\left((1 + \frac{\rho}{\alpha}) - e^{-j\psi}\right)}. \tag{3.29}$$

  The real and imaginary parts of the denominator on the right side of the above equation can be written respectively as

$$\Re(\text{den}) = \left(1 + \frac{\rho}{\alpha}\right)\cos\psi - \cos(2\psi), \tag{3.30}$$

$$\Im(\text{den}) = -\left(1 + \frac{\rho}{\alpha}\right)\sin\psi + \sin(2\psi). \tag{3.31}$$

  As α is real, the imaginary part must be zero, which requires

$$\cos\psi = 0.5\left(1 + \frac{\rho}{\alpha}\right). \tag{3.32}$$

  Combining the above results gives us

$$\Re(\text{den}) = 1, \tag{3.33}$$

and

$$\alpha = 0.5\left(1 \pm \sqrt{1-4\rho}\right). \tag{3.34}$$

For $\rho > 0.25$, there is no real-valued solution for α, and the ADPLL is unstable irrespective of α. For $\rho = 0.25$, we have one solution, $\alpha = 0.5$, based on the stability boundary condition, but the ADPLL is unstable for any other value of α. For $\rho < 0.25$, there are two real solutions for α as the boundary condition according to Eq. (3.34). The ADPLL is stable when α is between those two boundary values; otherwise unstable. In summary, the stability requirements dictate that $\rho < 0.25$ and $0.5\left(1-\sqrt{1-4\rho}\right) < \alpha < 0.5\left(1-\sqrt{1-4\rho}\right)$.

3.2.2.4 Phase transfer function

The z-domain frequency transfer function is convenient for basic analysis, including the stability and transient response characterization of the ADPLL. However, for characterizing the ADPLL output spectrum, a different ADPLL model is needed to relate the excess phase of the DCO output with that of the frequency reference, where the output phase signal should be treated as a continuous-time signal.

3.2.2.5 Modeling excess reference phase fluctuation

The reference phase in the time domain could generally be expressed as

$$\theta_R(t) = 2\pi f_R t + \Delta\theta_R(t), \tag{3.35}$$

where f_R represents the nominal reference frequency. When the reference signal is used as the clock for sampling the DCO digital control words and the DCO output phase, the excess phase fluctuation, $\Delta\theta_R(t)$, results in a fluctuation in the sampling time around the ideal values. The kth sampling occurs at the corresponding rising edge of the reference signal, i.e. at $\theta_R(t) = 2k\pi$. Thus, the kth sampling time can be given by

$$t_s[k] = kT_R - \frac{\Delta\theta_R[k]}{2\pi f_R}, \tag{3.36}$$

where $\Delta\theta_R[k]$ represents $\Delta\theta_R(t)$ at $t = t_s[k]$. Clearly, the phase fluctuation, $\Delta\theta_R[k]$, makes the actual kth sampling time deviate from the nominal time by an amount of

$$\Delta t_R[k] = -\frac{\Delta\theta_R[k]}{2\pi f_R}. \tag{3.37}$$

This timing deviation, also referred to as the absolute timing jitter [52], in turn induces a corresponding error in the PDC output, as illustrated in Fig. 3.7. The PDC output error due to the reference timing jitter can be found to be

$$\Delta\Phi_v[k] = f_{\mathrm{dco}}[k-1]\Delta t_R[k]. \tag{3.38}$$

It has been noted that the DCO output frequency is set at the previous sampling time; hence, $f_{\mathrm{dco}}[k-1]$ rather than $f_{\mathrm{dco}}[k]$ is used on the right-hand side of the above equation. During the steady-state operation, we can approximate the instantaneous frequency with $F_{tar}f_R$. Combining the above results yields

$$\Delta\Phi_v[k] = -\frac{F_{tar}}{2\pi}\Delta\theta_R[k] \quad \Leftrightarrow \quad \Delta\Phi_v(z) = -\frac{F_{tar}}{2\pi}\Delta\theta_R(z). \tag{3.39}$$

Therefore, the excess phase fluctuation of the frequency reference can be modeled by adding the corresponding error signal at the PDC output.

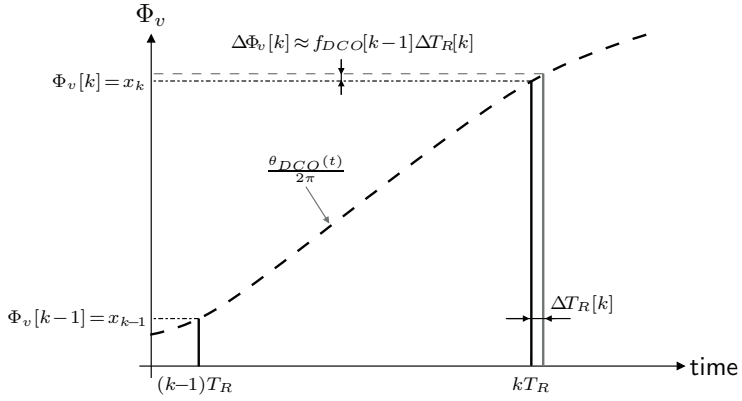


Figure 3.7. Effect of sampling time error on PDC output.

3.2.2.6 Frequency transform of the ADPLL output phase

In general, the frequency spectrum of a discrete-time signal represented in the z-domain can be obtained by evaluating the z-domain expression on the unit circle, i.e. $z = \exp(j2\pi f)$. This applies to the DCO output frequency when it is considered as a train of impulses in the discrete-time domain. However, the DCO output frequency is in fact a staircase-like waveform instead of a train of impulses due to the implicit zero-order hold operation in the digital input sampling. This zero-order hold operation

has the impulse response of a rectangular function as

$$h_{ZOH}(t) = \text{rect}\left(\frac{t}{T_\text{R}} - \frac{1}{2}\right) = \begin{cases} 1 & \text{if } 0 < t < T_\text{R} \\ 0 & \text{otherwise} \end{cases} \tag{3.40}$$

Its frequency domain response can be obtained via the Fourier transform to be

$$H_{ZOH}(f) = \mathcal{F}\left\{h_{ZOH}(t)\right\} = \frac{1 - e^{-j2\pi f T_\text{R}}}{j2\pi f} = e^{-j\pi f T_\text{R}} \left(\frac{\sin(\pi f T_\text{R})}{\pi f}\right). \tag{3.41}$$

The output phase is the integral of frequency in the continuous-time domain, as given by Eq. (3.3). The integral in time domain corresponds to a frequency response of $1/(2\pi f)$. The overall transfer function from the output frequency in the discrete-time domain to the output phase in the continuous-time domain can be written as

$$H_{out}(f) = \left(\frac{1}{j2\pi f}\right) H_{ZOH}(f) = \left(\frac{e^{-j\pi f T_\text{R}}}{j2\pi f^2}\right) \text{sinc}(\pi f T_\text{R}), \tag{3.42}$$

where $\text{sinc}(x)$ is the sinc function, i.e. $\text{sinc}(x) = \sin(x)/x$.

3.2.2.7 Complete ADPLL modeling for phase output

With the above results, the z-domain model of the ADPLL can be extended to include the reference phase input and the DCO phase output. The resulting mixed-domain model is shown in Fig. 3.8, where the target frequency, F_{tar}, defined by the ***FCW*** is considered a constant.

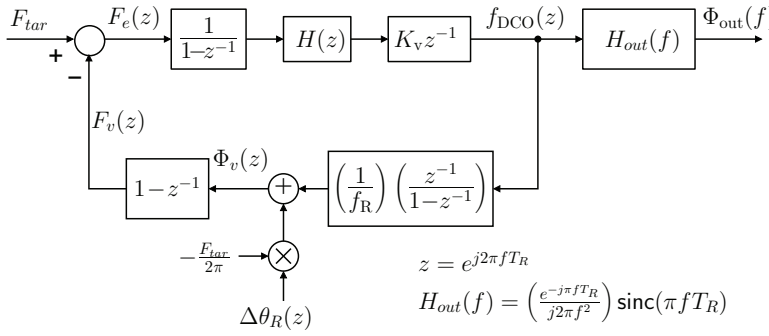


Figure 3.8. A mixed-domain model of the ADPLL.

Based on this ADPLL model, the transfer function from $\Delta\theta_R(z)$ to $\Phi_{out}(f)$ can be obtained as

$$G_\Phi(f) = \frac{\Phi_{out}(f)}{\theta_R(z)} = \left(\frac{F_{tar} f_R H(z) z^{-1}(1 - z^{-1})}{1 - z^{-1} + H(z) z^{-2}}\right) \left(\frac{e^{-j\pi f T_R}}{j(2\pi f)^2}\right) \text{sinc}(\pi f T_R), \tag{3.43}$$

where $z = \exp(j2\pi f)$. The DCO gain normalization has been assumed.

3.3 Performance analysis

To meet the same set of performance requirements associated with frequency synthesis, the ADPLL design shares some common considerations as with the conventional PLL design, but also entails some different considerations arising from its distinctive features in the architecture and operation principles. It is essential to examine the general considerations and identify the key challenges with the ADPLL design to meet performance requirements.

3.3.1 Frequency settling

Quantitative estimation of the ADPLL transient response can be investigated with the z-domain transfer function, Eq. (3.22), by using the inverse z-transform. The output frequency sequence, which is a response to a step change in the FCW, has a general expression of

$$f_{\text{dco}}(kT_R) = \mathcal{Z}^{-1}\left\{\frac{A_{st}G_f(z)}{1-z^{-1}}\right\} = A_{st}f_R \cdot \mathcal{Z}^{-1}\left\{\frac{G_{f,N}(z)}{1-z^{-1}}\right\}, \tag{3.44}$$

where $\mathcal{Z}^{-1}$ denotes the inverse z-transform. The term A_{st} in the above expression is the magnitude of the step change in the FCW value, F_{tar}, which means a change of $A_{st}f_R$ in the target frequency.

One observation is that the frequency settling given by Eq. (3.44) is independent of f_R if $A_{st}f_R$ is a constant irrespective of f_R, i.e. making the input step change for the same magnitude in Hertz in all cases. Meanwhile, since the time, kT_R, is proportional to the reference period, f_R, the frequency settling time also scales proportionally with the reference period or inversely with the reference frequency, f_R. However, the assumption is that the DCO gain is always normalized to f_R and the rest of the loop remains unchanged. This means the loop bandwidth also scales proportionally with the reference frequency. In practical implementations, the loop bandwidth usually needs to narrow correspondingly with an increased reference frequency, to adequately suppress on the high-frequency noise components, which would slow down the frequency settling and offset the settling speed gain from the increase in the reference frequency.

In the simplest case, where the loop filter is simply a gain constant, we have

$$f_{\mathrm{dco}}(z) = G_f(z) f_{\mathrm{tar}}(z) = \frac{\alpha f_R z^{-1}}{1 - z^{-1} + \alpha z^{-2}} f_{\mathrm{tar}}(z). \tag{3.45}$$

Using the step function $f_{\mathrm{tar}}(z) = 1/\left(f_R(1 - z^{-1})\right)$, the output frequency can be written as

$$f_{\mathrm{dco}}(z) = \frac{\alpha z^{-1}}{(1 - z^{-1})(1 - z^{-1} + \alpha z^{-2})}; \tag{3.46}$$

its inverse z-transform then yields

$$\begin{aligned} f_{\mathrm{dco}}[k] =& 1 - \frac{1}{2\sqrt{1-4\alpha}}(2^{-k}((1+\sqrt{1-4\alpha})^k(1+\sqrt{1-4\alpha}-2\alpha) \\ &+ (1-\sqrt{1-4\alpha})^k(-1+\sqrt{1-4\alpha}+2\alpha))). \end{aligned} \tag{3.47}$$

Some examples of the ADPLL transient response with different values of α are plotted in Fig. 3.9.

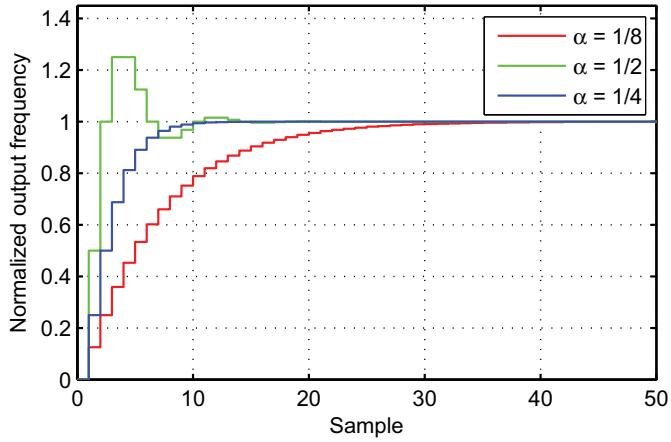


Figure 3.9. ADPLL transient response from inverse z-transform.

In a digital PLL, it is common to employ circuit techniques, such as adaptive frequency calibration (AFC), for frequency coarse tuning prior to frequency fine tuning and tracking to speed up the frequency switching. The ADPLL architecture, which has the frequency error information available in the digital domain, lends itself particularly well to these techniques. However, the key for fast frequency settling lies with the PDC resolution in the feedback path. It determines how much techniques like AFC can ultimately speed up the settling process. It also sets a limit on the loop bandwidth for fine tuning and frequency tracking, which in turn limits the frequency settling speed in the fine tuning. A fine PDC resolution is

thus essential to ensure a short settling time.

3.3.2 Phase noise and spurious tones

The major noise sources in an ADPLL are associated with the DCO and the PDC. With due approximations, these noise sources can be included in the ADPLL model, as in Fig. 3.8, to account for their effects on the output spectrum. Next, we summarize relevant results that have been presented in the existing literature [4, 53–55].

As in a typical PLL, the in-loop tunable oscillator naturally remains a dominant phase noise source. It is the purpose of the negative feedback mechanism of a PLL to stabilize this oscillator and suppress its close-in phase noise. For a PLL to achieve a low phase noise level, the tunable oscillator itself should have good phase noise performance and the loop should provide adequate noise suppression with a sufficient loop bandwidth. The implication of a digital interface for the oscillator in an ADPLL is twofold in terms of the phase noise performance. By using digital tuning instead of analog, the oscillator output frequency can be made relatively insensitive to the voltage fluctuation of the control lines, which helps to reduce the close-in phase noise level. On the other hand, the digital interface is typically associated with a relatively large number of devices with an increased amount of wiring, which can potentially lead to higher loss and increased phase noise. More importantly, digital frequency tuning means a finite frequency resolution, with the DCO output frequency is not tuned continuously but in steps. The frequency quantization is a source of nonlinearity in the loop dynamics, which tends to degrade the overall phase noise performance and give rise to spurious tones in the ADPLL output. Predicting spurious tones is usually difficult with mathematical tools. The phase noise contribution from this frequency quantization can be quantified with a linear approximation in which quantization errors are assumed to be white noises. A previous study [54] formulated this phase noise contribution as

$$\mathcal{L}\{f_m\} = \frac{1}{12} \cdot \left(\frac{\Delta f_{\text{dco}}}{f_m}\right)^2 \cdot \frac{1}{f_R} \cdot \left(\text{sinc}\frac{f_m}{f_R}\right)^2, \tag{3.48}$$

where f_m represents the offset frequency, Δf_{dco} the DCO frequency resolution and f_R the reference frequency. This result, however, ignored the noise-folding effect of the sub-sampling operation in the feedback path.

Consequently, it does not account for the in-band noise contribution. As later elaborated in another study [55], the in-band noise contribution of the DCO frequency quantization can be approximated as,

$$\mathcal{L} = \left(2\pi \cdot \frac{\Delta f_{\text{dco}}}{f_R}\right)^2 \cdot \frac{1}{12 f_R}. \tag{3.49}$$

To minimize the effect of the DCO frequency quantization, the key is to have a fine DCO frequency resolution. In nanoscale CMOS technologies and with appropriate circuit techniques, the frequency quantization can be made sufficiently low for typical wireless applications [3, 9].

Another important source of phase noises is related to the finite resolution of the phase digitization operation in the feedback path. This phase digitization usually employs a time-to-digital converter (TDC) as its core functional block; the time resolution of the TDC is subject to a number of practical limitations. The time quantization errors of the TDC could dominate the in-band phase noise at the ADPLL output and they also limit the viable loop bandwidth. With the linear approximation, the contribution of this phase noise digitization can be quantified as follows [53]

$$\mathcal{L} = \frac{1}{12 f_R} \left(2\pi f_{\text{dco}} \Delta t_{\text{tdc}}\right)^2 = \frac{1}{12 f_R} \left(\frac{2\pi \Delta t_{\text{tdc}}}{T_{\text{dco}}}\right)^2, \tag{3.50}$$

where Δt_{tdc} is the TDC time resolution and T_{dco} represents the period of the DCO output.

The limitation on the loop bandwidth imposed by the feedback quantization reduces the suppression of DCO phase noises and thus limits the overall phase noise performance.

3.3.3 Frequency tuning range

The frequency tuning range of an ADPLL is mainly determined by the DCO in the loop. Though it can be extended beyond that of the DCO with additional circuit blocks such as frequency dividers, there are several limitations with this extension. In general, only power-of-two frequency dividers are practically feasible or cost-effective for this purpose, which means the maximum divided-down frequency is at best half the maximum frequency prior to the frequency division. This would result in only disjoint frequency sub-ranges instead of a continual one if the oscillator itself cannot cover an octave frequency band characterized with a frequency

ratio (f_{max}/f_{min}) of two. In addition, the need for multiplexing also to a certain degree complicates the frequency-division approach of keeping the coupling of different branches to an acceptable level [56]. Designing a high-frequency CMOS oscillator to cover an octave band is certainly not a trivial task; it involves trade-offs with other important performance parameters, such as the phase noise, power consumption and silicon area.

The frequency tuning range also bears design implications for the PDC in the feedback path in different ways concerning the PDC resolution, power and silicon area. To handle the RF signal over a wide frequency range, the PDC could face conflicting design scenarios associated with the frequencies at both ends of the frequency range.

3.3.4 Frequency resolution

As with a conventional digital PLL, the frequency resolution of an ADPLL is determined by the number of fractional bits in the *FCW* and the reference frequency. It can be expressed as

$$\Delta f = \frac{1}{2^{W_F}} \cdot f_R, \tag{3.51}$$

where W_F and f_R are the number of FCW fractional bits and the reference frequency, respectively. Basically, there is no fundamental limitation on the number of fractional bits and thus the frequency resolution, while the digital implementation for a finer resolution could be more complex due to the associated larger word length. As for a conventional PLL, techniques such as $\Sigma\Delta$-modulation can also be employed to shorten the effective FCW length before it is used for phase or frequency error detection, which allows the subsequent digital blocks to implemented with a shorter word length. The result is that a very fine frequency resolution can be achieved at little cost in terms of power and silicon area. Note that the ADPLL frequency resolution should be distinguished from the DCO frequency resolution, which are independent of each other.

3.4 ADPLL architecture alternatives

The ADPLL architecture can be rearranged to arrive at a slightly different version, as shown in Fig. 3.10. In this architecture version, the accumulation of the *FCW* for each reference cycle constitutes the reference phase

signal. The digital DCO phase information from the PDC is subtracted directly from the reference phase to form the phase error, which is conditioned with a digital loop filter (DLF) for DCO frequency tuning. This architecture version can be viewed as result of moving the frequency error accumulator in the architecture of Fig. 3.1 back to the input branches of the frequency error detector and canceling out the difference in the feedback path. Elaboration of this architecture version can be found in an earlier study by Staszewski [49]. Its operation is quite similar. However, a major disadvantage of this architecture version involves the loss of important DCO frequency information in the digital domain, which does not make it so appealing.

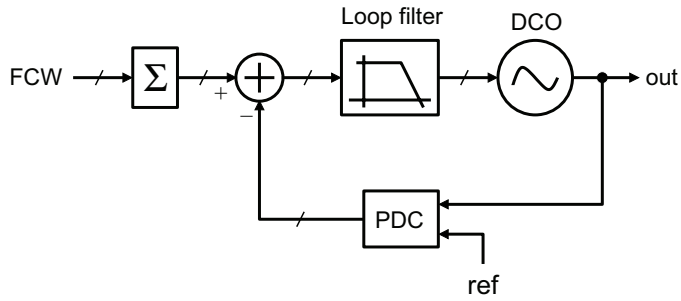


Figure 3.10. Phase-domain ADPLL architecture using phase comparison.

An ADPLL can also be realized by simply replacing the analog blocks in a conventional charge-pump PLL with their digital counterparts. The resulting architecture is illustrated in Fig. 3.11. Specifically, the charge pump, along with the PFD, is replaced with a TDC, which converts the phase error between the divided-down output signal and the frequency reference signal into its digital presentation. By digitizing the phase error with the TDC, the following low-pass loop filter (LPF) can then be implemented in the digital domain. The DCO is a counterpart to the VCO in conventional PLLs, allowing digital control of its output frequency. This architecture is conceptually straightforward compared to that of a conventional PLL. As in a convention PLL, the frequency control in this architecture is performed by defining the division ratio of the multi-modulus frequency divider (MMD) in the feedback path. An example of this architecture can be found in a study by Hsu et al [57], where the DCO is implemented as a combination of a digital-to-analog (D/A) converter and a VCO, and the TDC employs a gated-ring-oscillator (GRO) structure.

The similarity of this architecture to a conventional charge-pump architecture allows it to better leverage existing theories and techniques de-

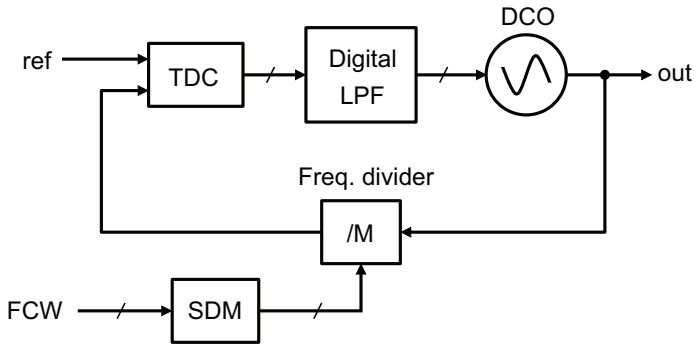


Figure 3.11. Phase-error-digitized ADPLL architecture.

veloped for the conventional architecture over a relatively long period of time. However, it also inherits from the conventional PLLs the same design issues and challenges associated with the feedback path and the SDM-based fractional frequency control approach. Particularly, the design of an MMD with a large division ratio range has difficulty handling the RF signal, and its speed and power consumption constitute a potential performance bottleneck for the ADPLL. In addition, realizing effective fractional division ratio by means of switching back and forth among integer ratios, the feedback is known to be the source of substantial quantization noises. This noise source does not exist in the architecture shown in Fig. 3.1. Although digital cancellation techniques can be employed to mitigate the issue [57], they also increase the design complexity.

In view of relative drawbacks of the above architecture variants, they have not been the focus of this work. Instead, the architecture shown in Fig. 3.1 remains the architecture of choice throughout this study. It has been the basis for further architectural improvements and circuit-level techniques developed in the author's work and is going to be the focus of further discussion in next chapters.

3.5 Summary

An ADPLL comprises an DCO for RF frequency generation and a digital-intensive feedback loop to control its output frequency. There are a few general architectures that feature different loop configurations. The popular approach is to convert the RF output phase into a digital form in the feedback path and process the phase signal in the digital domain to generate a corresponding digital result for the DCO frequency control. Despite

all the previously mentioned benefits, the digital approach is also associated with some obvious disadvantages. It usually incurs more loop delay than its analog counterpart due to the need for sampling at the DCO input to align different bits in the control word. In addition, quantization errors are also inherent with the digital implementation.

The basic function of an ADPLL can be modeled in the discrete-time domain and its z-transform. However, when characterizing its output spectrum requires, the output phase needs to be considered as a continuous-time signal, which makes the resulting model relatively complex.

4. Frequency acquisition

A PLL settling process can be considered one of lock acquisition, which brings the loop from an initial condition into lock. In wireless applications, fast acquisition is always desired from system perspective. In general, it starts with frequency acquisition and progresses to phase acquisition before lock is achieved. For fast settling, both frequency acquisition and phase acquisition need to be optimized. In particular, frequency acquisition tends to be the key in determining the settling speed [51].

This chapter begins with by discussing the general principles of lock acquisition within the framework of an ADPLL and then describes different techniques for achieving it. It highlights a fast frequency acquisition technique that makes it possible for an ADPLL to settle at a maximized speed.

4.1 Overview of ADPLL acquisition

4.1.1 Review of ADPLL lock state

As is known from previous analysis, the detected frequency error in the digital domain, with the quantization error neglected, can be given by a difference equation as

$$F_e[n] = F_{tar} - \frac{1}{f_R}\left(f_0 + K_{DCO}d[n-1]\right), \tag{4.1}$$

where $d[n-1]$ is the value of the DCO control word at the previous reference clock cycle and f_0 is the DCO frequency when the control word is zero. The phase error is derived through accumulating the frequency

error and can be expressed as

$$\Phi_e[n] = \Phi_e[0] + \sum_{k=1}^{n} F_e[k]. \tag{4.2}$$

In a phase-locked state, the phase error by definition should be constant, or $\Phi_e[n] = \Phi_e[n-1]$ for any reference cycle index, n, over the relevant period of time. This requires that $F_e[n] = 0$ for any relevant value of the cycle index, n. Conversely, if $F_e[n] = 0$ for any relevant value of n, it follows from Eq. (4.2) that $\Phi_e[n] = \Phi_e[n-1]$, i.e. the phase error is constant and the ADPLL is phase locked. This basically says that the ADPLL can be automatically locked when the frequency acquisition is fully performed. It should be noted that this is a special feature of an ADPLL and one that is associated with the unique method of phase error detection. It is generally not true with a conventional PLL, where the phase detection method dictates that phase acquisition is still needed even if frequency acquisition is performed perfectly. That need arises by the fact that even if the PLL output frequency is exactly the target frequency, there is still the same possibility of phase misalignment between the frequency-divided feedback signal and the reference clock. With an ADPLL, there is no such frequency-divided signal and its associated phase alignment with the reference clock. As a result, the additional phase acquisition process is not needed, which in principle allows an ADPLL to potentially settle faster than a conventional PLL.

The question that has not been answered yet has to do with the lock state of the ADPLL if we look at one point in time or one single reference clock cycle. Clearly, a zero frequency error for one cycle does not guarantee that it would remain so, and it thus does not tell us whether or not the loop is in a locked state. To address such a question, we need to take the whole loop into account. Particularly, the DCO input is the output of the loop filter, and it can in general be given by,

$$d[n] = \sum_{i=1}^{N} a_i d[n-i] + \sum_{j=0}^{M} b_j \Phi_e[n-j], \tag{4.3}$$

with a_i and b_j being the loop filter coefficients. The terms $d[n-i]$ and $\Phi_e[n-j]$ are respectively delayed versions of $d[n]$ and $\Phi_e[n]$, and they are stored in one form or another in the memory elements (not always separately) as state variables of the loop filter. In the simplest case of a type-I ADPLL, the loop filter degenerates to a gain constant, and we have

$d[n] = G_{lpf}\Phi_e[n]$ with no additional state variables involved. It can be shown that a PI loop filter for a type-II ADPLL can generally be described as

$$d[n] = d[n-1] + \left(\left(K_P + K_I\right)\Phi_e[n] - K_P\Phi_e[n-1]\right).$$

To better reflect a typical circuit implementation, the difference equation can be rewritten as

$$d[n] = K_P\Phi_e[n] + \left(K_I\Phi_e[n] + d_I[n-1]\right),$$

where

$$d_I[n-1] = d[n-1] - K_P\Phi_e[n-1]$$

is the delayed version of the integrator branch output and usually stored as one state variable.

When an ADPLL is correctly locked, the phase error and DCO input should remain constant. As a result, it follows from Eq. (4.3) that

$$d[n] = G_{lpf} \cdot \Phi_e[n] \;\text{ or }\; \Phi_e[n] = \frac{d[n]}{G_{lpf}}, \tag{4.4}$$

with $G_{lpf} = \sum_{j=0}^{M} b_j / \left(1 - \sum_{i=1}^{N} a_i\right)$ being the loop filter DC gain. The above relationship holds true regardless of the index n value as long as the ADPLL is in a locked state. It basically says that the DCO input is equal to phase error scaled by the loop filter DC gain. Since $F_e[n] = 0$ in a locked state, it follows from Eq. (4.1) and Eq. (4.4) that

$$d[n] = \frac{f_{tar} - f_0}{K_{DCO}}, \;\text{ or }\; \Phi_e[n] = \frac{f_{tar} - f_0}{G_{lpf}K_{DCO}}. \tag{4.5}$$

It should be noted that the loop filter DC gain, G_{lpf}, can be an infinity due to the presence of one or more integrators, which would force $\Phi_e[n]$ to be zero in a steady locked state.

In summary, for the ADPLL to be in a steady locked state, the control word, $d[n]$, and phase error, $\Phi_e[n]$, along with their delayed versions, $d[n-i]$ and $\Phi_e[n-j]$, in Eq. (4.3) acting as state variables of the loop filter, should be at their respective steady-state values according to Eq. (4.5). With a stable ADPLL design, we can claim that the ADPLL is in a phase-locked state, if the above is true. That is, if $d[n]$ and $\Phi_e[n]$ along with the loop filter state variables are at the steady-state values in the nth cycle,then their values will be sustained with the DCO continuously generating an RF

output exactly at the target frequency. However, with an unstable ADPLL design, the disturbance from the noises and quantization errors would drive the ADPLL away from the expected lock state and lock becomes impossible.

4.1.2 ADPLL frequency acquisition overview

In the ADPLL tracking mode, the phase digitization of the DCO output signal needs to be based on a TDC to provide a fine resolution that can minimize the level of quantization errors. The TDC-based phase digitization is in general characterized by a measurement range of one RF cycle or 2π. As a result, the ADPLL output frequency needs to be within the range of $\pm f_R/2$ from the target frequency in order to ensure the frequency error detection is performed correctly in the tracking mode. Meanwhile, the capacitor bank of a DCO used for frequency tracking, along with the need to provide a fine frequency resolution to minimize frequency quantization, is usually able to cover only a limited frequency range. These limitations dictate that the frequency tracking loop is not capable of full frequency acquisition and that additional circuitry is needed to aid the acquisition.

It should be noted that acquisition-aiding circuitry is also commonly used for a conventional PLL, where similar limitations also exist. Phase error detection in the tracking mode of a conventional PLL is typically characterized by a small operating range and thus is not suitable for frequency acquisition. A conventional VCO also needs to be designed for a limited tracking frequency range with a moderate frequency gain in order to keep down noise contribution from the voltage control line. By comparison, a VCO can usually be designed to have a tracking range somewhat larger than that of a DCO, as the component matching and physical size of the capacitor bank tends to impose a harder restriction than the consideration of noise upconversion.

With a different functionality, the acquisition circuitry for a PLL can be designed in a way relatively independent of the tracking loop. While a conventional PLL design is overall analog-intensive, a relatively digital approach is often employed for frequency acquisition circuitry. It is common that the VCO for a conventional PLL is designed to have a capacitor bank that is digitally controlled during frequency acquisition [58]. As a result, many of the frequency acquisition techniques suitable for conven-

tional PLLs can in principle also be used with an ADPLL. However, it should be noted that the requirement for frequency acquisition resolution tends to be higher. This is because of the more limited DCO tracking frequency range as compared to that of a VCO.

4.2 Frequency calibration techniques

The frequency acquisition of a digital PLL is also a process of calibrating the DCO frequency so that it can operate properly in subsequent frequency tracking. Next, we review different frequency calibration techniques that have been proposed in the literature for fast frequency acquisition. Among the techniques that will be discussed, the author has applied the PLL-based frequency calibration in one ADPLL design [12], and a technique of binary search approximation in other designs [8, 13]. In particular, the author is the first to propose a dynamic binary search for an ADPLL frequency synthesizer [13].

4.2.1 PLL-based frequency calibration

In previous studies by Staszewski and others [59–62], apparently the same kind of loop for frequency tracking with phase error detection and filtering has also been used for frequency acquisition. In the feedback path, a counter-based phase accumulator is employed to digitize the DCO phase in the acquisition mode in order to allow for a large conversion range. At the loop filter output, mode-switching logic allows for the progression of an active frequency control from one capacitor bank of the DCO to another. A major disadvantage of this approach is the difficulty in having accurate timing control for the mode transition. Its performance is subject to the effect of PVT variations of the DCO gain, which results variations in the PLL loop bandwidth. The lack of a clear indicator regarding when active control would be transferred from one bank to the next could result in either a waste of time or acquisition failure. Meanwhile, sharing the loop filter in different modes can actually increase the complexity of the loop filter and complicate the overall design. The need for filtering is supposed to vary dramatically over different modes according to the level of quantization error and the changing requirement for the instantaneous frequency resolution over the entire acquisition process. To accommodate the varying requirement, the loop filter needs to be

designed with increased word lengths and more configurable parameters. Meanwhile, it incurs additional multiplexing and state variable resetting logic during the mode transitions. All of the hardware overhead tends to outweigh the potential benefit resulting from the hardware sharing. Meanwhile, it has been pointed out previously that the claim about the joint operation of the phase accumulator and TDC for phase digitization is merely a misunderstanding. Basically, the TDC is ineffective in the acquisition mode whether it is active or not, while the counter-based phase accumulator is not needed in the tracking mode.

4.2.2 Code estimation and presetting

DCO code presetting or forward compensation techniques are based on an estimation of the DCO parameters for presetting the DCO according to the target frequency before the frequency settling [63–65]. Such techniques are typically based on the assumption of a practically linear DCO. As shown in Fig 4.1, the transfer function of a linear DCO can be represented by a straight line that is unambiguously defined with the knowledge of its two points, particularly the two end points. More specifically, suppose $f_{\min}$ is the frequency for a minimum DCO control code, $d_{\min}$, while $f_{\max}$ is the frequency for a maximum DCO control code, $d_{\max}$. It follows that the DCO output frequency, f_x, for an arbitrary DCO control code, d_x, would be

$$f_x = f_{\min} + \frac{f_{\min} - f_{\min}}{d_{\max} - d_{\min}} (d_x - d_{\min}) . \tag{4.6}$$

In other words, if the target frequency is f_x, the DCO control code can be ideally set to be

$$d_x = d_{\min} + \frac{d_{\max} - d_{\min}}{f_{\min} - f_{\min}} (f_x - f_{\min}) . \tag{4.7}$$

In practice, the numbers will need to be rounded off because the control code should consist of whole numbers. The DCO output frequency is not known directly. Instead, its quantized version at the output of the feedback path can be used in the above estimation or it can be derived or approximated with other signal values [63]; however the latter option is a less straightforward and accurate option. The information about the target frequency, f_x, is in the FCW to the ADPLL.

It should be noted that the adoption of this type of techniques has been limited to relatively low-performance applications, where the requirement

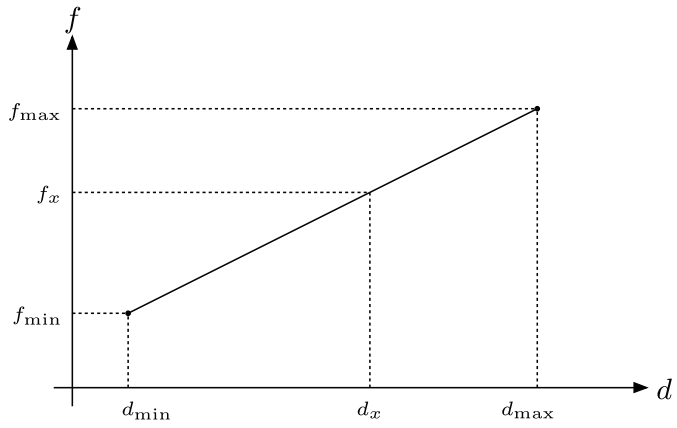


Figure 4.1. Transfer characteristics of a linear DCO.

for the phase noise and frequency resolution is low. The major issue with the techniques is that the achievable accuracy is typically very coarse in comparison. This is due to various factors. The most important of these factors are:

- First, the overall transfer function of a DCO can be quite nonlinear, and thus a linear assumption is usually not justified for the whole tuning range. This is particularly true with an LC DCO, whose oscillation frequency is a nonlinear function of the inductance and capacitance of its LC tank. Meanwhile, whether it is an LC oscillator or a ring oscillator, the component mismatches of the tuning bank limit the linearity of a DCO.

- Second, the quantization of the DCO output frequency is subject to the limited resolution of the feedback path.

Another issue with the techniques, one which Eq. (4.7) demonstrates is that estimation of the control code usually involves arithmetic division or multiplication, which tends to make it costly for hardware implementation. Moreover, there are typically multiple frequency tuning banks instead of a single one in a DCO. As a result, multiple DCO control codes need to be estimated, which further complicates their implementation. Among the multiple tuning banks, the quantization step of a coarse bank tends to be one or two orders of magnitude larger than that of the next finer bank. In other words, one quantization step of the coarse bank can

typically be on the same order of magnitude as the whole tuning range of the next finer bank. As a result, any small inaccuracy in presetting the coarse code, which is almost inevitable, tends to invalidate the presetting for finer banks.

4.2.3 Binary search successive approximation

Successive approximation frequency calibrations, typically based on a binary search (BS) algorithm, have been widely used to aid frequency acquisition of different PLL architectures [8, 66–69]. Their popularity has to do with their simplicity, robustness and relatively high speed. The basic algorithm of a BS-based calibration is depicted in Fig. 4.2. It works by adjusting the DCO frequency through its control word on a step-by-step basis with the frequency step size halved for each iteration. It starts by setting the DCO control word and the step size to their appropriate initial values. For each step, a frequency comparison is performed between the output frequency and the target frequency. Based on the comparison result, the control word may then be changed to step up or down the output frequency. The iteration goes on until the last step or the target frequency is reached.

In general, the frequency comparison can be performed only with a finite resolution that is limited by the time allowed for the frequency measurement. The BS calibration resolution is in turn limited by the frequency comparison resolution. When the difference between the output frequency and the target frequency is smaller than the frequency comparison resolution, the comparison cannot distinguish which frequency is higher or lower and thus the output frequency cannot be reliably brought closer to the target frequency. With a typical BS calibration, the comparison resolution is a design parameter fixed throughout the whole binary search process. Improving the comparison resolution tends to require a proportional increase in the frequency measurement time for each step. As a result, fast frequency acquisition usually comes with a relatively poor calibration resolution, and a significant frequency settling time is still needed after the calibration phase, especially for applications with small frequency tolerances. Attempts to improve the calibration resolution tend to introduce proportional time increase for each search step [69], which dramatically diminishes the advantage of the calibration speed.

The performance of a BS calibration can be improved with a dynamic

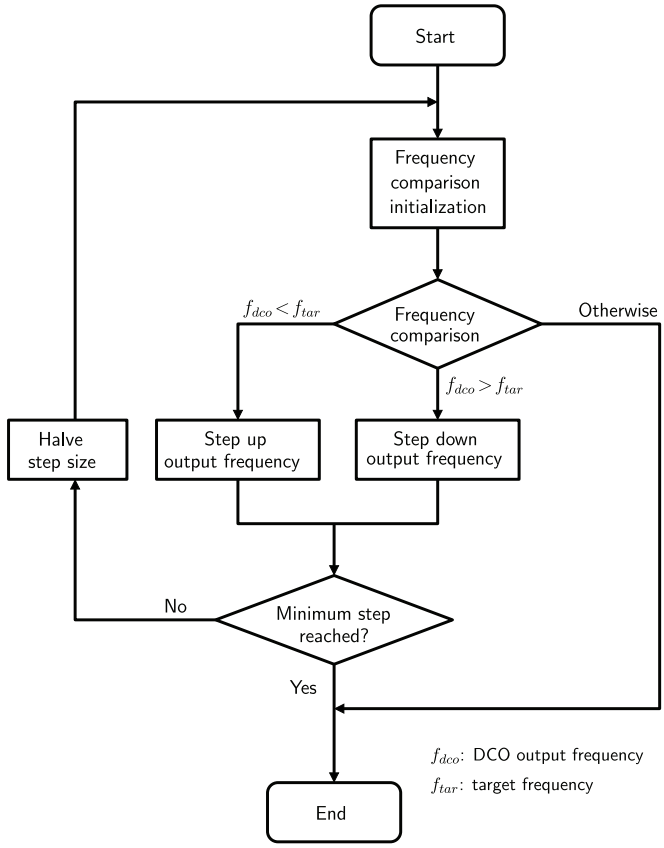


Figure 4.2. Basic binary search frequency calibration.

resolution setting for each search step, as proposed for the ADPLL in a published study by the author [13]. The principle of such a dynamic BS calibration is depicted in Fig. 4.3. It features a dynamic frequency resolution improvement part in addition to what is common to a traditional BS calibration technique. At each search step, the frequency comparison is first performed with an initial resolution. If the frequency difference is too small for comparison, the resolution is refined for further comparison at the expense of more frequency measurement time until the final resolution is reached. Such a technique makes it possible to use a relatively coarse comparison resolution for most of the search steps without sacrificing the overall calibration resolution. As a result, it allows for much faster calibration for a given calibration resolution.

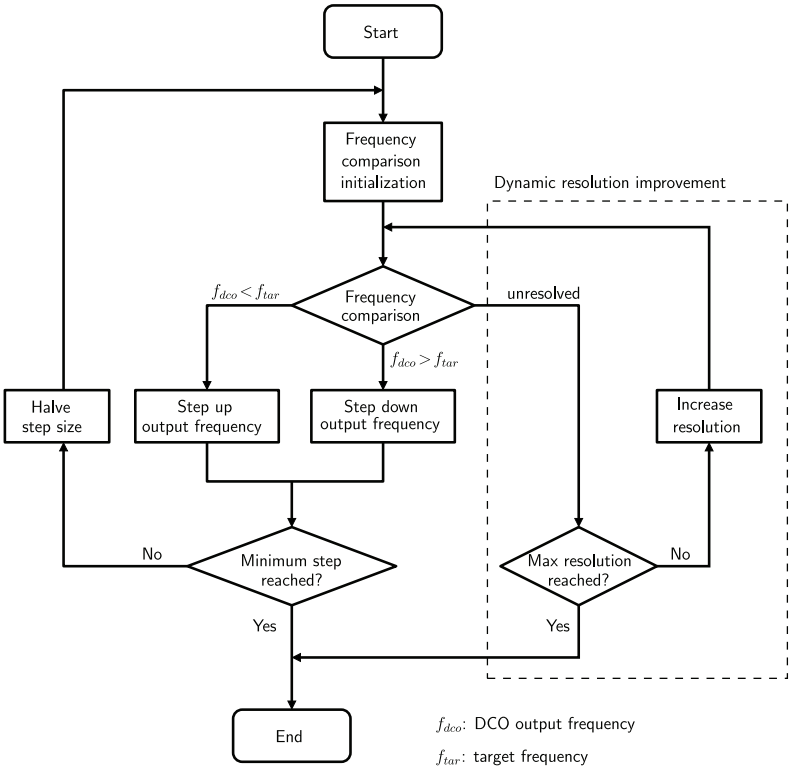


Figure 4.3. Dynamic binary search frequency calibration.

4.3 Summary and discussion

In this section, the general features of the ADPLL acquisition process have been reviewed. It was noted that the additional phase acquisition, which is typically an integral part of the whole acquisition process for a conventional PLL, is no longer necessary with the ADPLL architecture. With the ADPLL, DCO frequency calibration is a key part of frequency acquisition. Different calibration techniques were described and summarized here. Among the different calibration techniques, the BS-based successive approximation features with superior robustness and simplicity. In particular, the dynamic binary search first proposed by the author [13] for an ADPLL also allows for fast calibration with a fine resolution, which is essential for ensuring the rapid settling of an ADPLL frequency synthesizer.

5. Digitally controlled oscillator

A PLL-based frequency synthesizer commonly employs one or more RF oscillators to generate its output signal waveform. In an ADPLL, the RF oscillator is designed to have a digital interface for frequency control. The digitally controlled oscillator (DCO) in an ADPLL is functionally a digital-to-frequency (D/F) converter. From this perspective, it is a special same digital-to-analog (D/A) converter and shares the D/A converter design considerations, such as linearity and resolution. From another perspective, the DCO in an ADPLL plays the same role as a voltage-controlled oscillator (VCO) in a conventional PLL. Except for the interface, the RF oscillator core of a DCO is essentially the same as that of a VCO. As a result, it shares key design considerations with the latter. In our own work, we have designed

- an LC DCO [9], which was later applied in a 2.4GHz ADPLL design [12],
- a ring DCO with a band-extension LC tank [11], which was integrated with a wideband ADPLL for a cognitive radio application [8], and
- a two-stage ring DCO utilized in another wideband ADPLL design [13].

This section examines DCO design principles from both perspectives, while details of my own DCO designs are presented in Chapter 7.

5.1 Oscillator fundamentals

5.1.1 Oscillator models

This section examines the principles of oscillator frequency tuning and considers digital frequency tuning with a DCO in its proper perspective in order to facilitate further discussion. More comprehensive coverage of

oscillator theories can be found in numerous published works [70–77].

An oscillator generates a periodic signal waveform when powered from a DC supply. With CMOS technology, there are primarily two types of oscillators: LC oscillators and ring oscillators. An LC oscillator is basically a combination of an LC resonator with an active network. A ring oscillator is built with a number of delay stages, with the output from the last stage fed back to the input of the first stage to form a ring. Examples of ring oscillators and LC oscillators are given in Fig. 5.1 and Fig. 5.2.

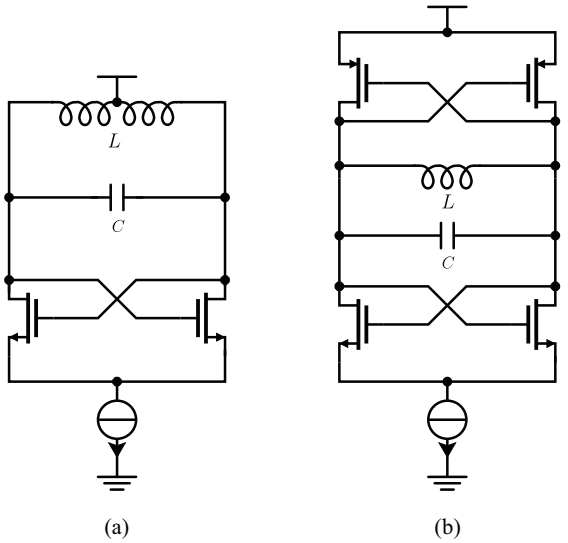


Figure 5.1. LC oscillators: (a) NMOS-core, and (b) CMOS-core.

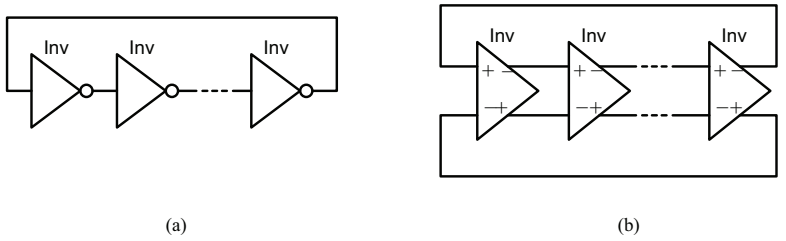


Figure 5.2. Ring oscillators: (a) single-ended, and (b) differential.

While it is inherently nonlinear, an oscillator can usually be represented by its linearized small-signal models to gain insight into its operation. A common model is a positive feedback system, as shown in Fig. 5.3, which consists of a gain block $G(j\omega)$ and a frequency selective block $H(j\omega)$. The transfer function can be given as,

$$\frac{V_{\text{out}}(j\omega)}{V_{in}(j\omega)} = \frac{G(j\omega)}{1 - G(j\omega) \cdot H(j\omega)}. \tag{5.1}$$

Sustainable oscillation means that the system can have non-zero output with zero input, which is true only if the denominator of the above transfer function is zero. In other words, we need the open-loop transfer function to satisfy the following condition,

$$G(j\omega) \cdot H(j\omega) = 1. \tag{5.2}$$

This is commonly known as the Barkhausen criterion. It can be interpreted as meaning that at the oscillation frequency, we have the open-loop gain magnitude, $|G(j\omega) \cdot H(j\omega)| = 1$, and its phase, $\angle(G(j\omega) \cdot H(j\omega)) = 0°$.

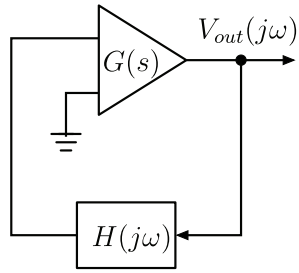


Figure 5.3. An oscillator model as a positive feedback system.

In an LC oscillator, the frequency selective block is the LC tank, while the gain block can be represented as a transconductance of the transistors. As a result, a single-ended linear model for an LC oscillator can be formed, as shown in Fig. 5.4, where L and C are respectively the tank inductance and capacitance, and the resistor, R_L, the tank loss. The input, I_{in}, is the model for the noise current or initial condition when the oscillator is powered up. The oscillation frequency can be derived from the Barkharsen criterion as

$$f_{\text{osc}} = \frac{1}{2\pi\sqrt{LC}}, \tag{5.3}$$

which is the resonant frequency of the LC tank.

In a ring oscillator, each inverter stage can be modeled as an RC network loading a transcondunctance amplifier via phase inversion. Fig. 5.5 shows a simplified linear model of an N-stage ring oscillator. Consequently, the open-loop transfer function can be written as,

$$H(j\omega) = \left(\frac{-G_m R}{1 + j\omega RC}\right)^N, \tag{5.4}$$

with N being an odd number of stages. It follows that at the oscillation

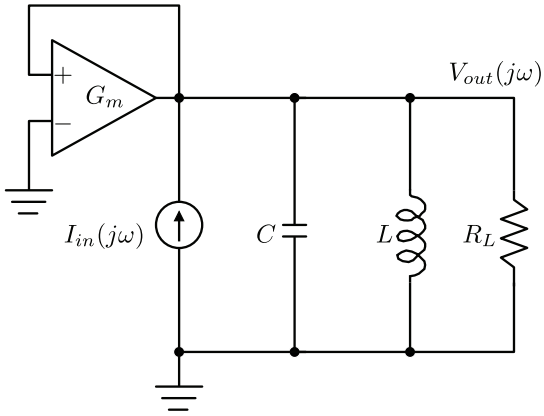


Figure 5.4. A linear model of an LC oscillator.

frequency, we have

$$\tan^{-1}(\omega RC) = \frac{\pi}{N}. \tag{5.5}$$

In other words, the oscillation frequency of an N-stage ring oscillator can be given as

$$f_{\text{osc}} = \frac{\tan(\pi/N)}{2\pi RC}. \tag{5.6}$$

The result shows that oscillation frequency of a ring oscillator can be determined by the number of stages and the amount of delay in each stage, as represented by the RC product.

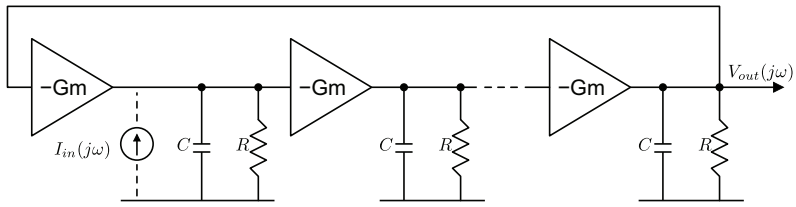


Figure 5.5. A linear model of a ring oscillator.

One assumption made above is that every stage is identical for a ring oscillator. In a more general case, a ring oscillator comprises non-identical stages and the open-loop transfer function can be written as

$$H(j\omega) = \prod_{i=1}^{N} \left(\frac{-G_{m,i}R_i}{1 + j\omega R_i C_i} \right), \tag{5.7}$$

while the oscillation frequency should be a solution of the following equation,

$$\sum_{i=1}^{N} \tan(\omega R_i C_i) = \pi. \tag{5.8}$$

A hybrid of an LC ring oscillator can be formed if an LC tank is introduced to a ring oscillator [11]. A linear model of such a hybrid LC ring oscillator is shown in Fig. 5.6. The inclusion of the LC tank can be controlled by a switch. When the LC tank is disconnected from the ring oscillator, the circuit works like a normal ring oscillator. When the LC tank is connected, its inductive impedance boosts the oscillation frequency of the ring oscillator. As a result, the frequency range can be extended upward beyond a typical ring oscillator. The open-loop transfer function, with the inclusion of an LC tank, can be written as

$$H(j\omega) = \frac{-G_m R}{1+jR\left(\omega C+\omega C_p-\frac{1}{\omega L_p}\right)} \cdot \left(\frac{-G_m R}{1+j\omega RC}\right)^2. \tag{5.9}$$

The new oscillation frequency, ω_{ON}, can be estimated as a solution to the following equation,

$$\tan^{-1}\left[R\left(\omega C+\omega C_p-\frac{1}{\omega L_p}\right)\right] + 2\tan^{-1}(\omega RC) = \pi. \tag{5.10}$$

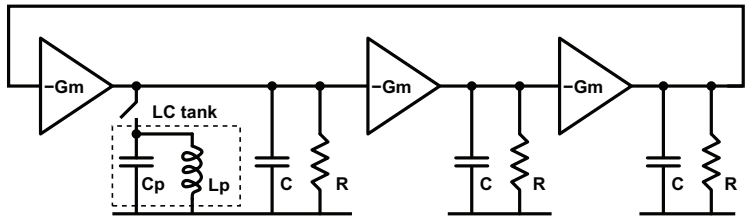


Figure 5.6. A linear model of the ring oscillator with an LC tank.

5.1.2 Oscillator performance measures

The commonly used metrics to characterize a DCO include the following:

- Frequency range: oscillators are often tunable over to cover a certain range of frequencies. The oscillator's frequency range or tuning range is often given in terms the minimum and maximum frequency and sometimes as the difference between the maximum and minimum frequencies.

- Tuning ratio: oscillator tuning ratio often refers to the ratio of difference between the maximum and minimum frequencies to the center frequency in terms of percentage. In other words, it is given as

$2 \cdot (f_{max} - f_{min}) / (f_{max} + f_{min})$. Sometimes, it is also given as the ratio of the maximum frequency to the minimum frequency of the frequency range.

- Frequency gain: with traditional VCOs, frequency gain refers to the frequency increase per unit voltage increase, and it is usually given in units of MHz/V. With DCOs, it typically refers to the frequency increase per one LSB increase in the control code value, and it is often given in units of MHz/V.

- Frequency resolution: frequency resolution is a performance metric used only for DCOs. It refers to the minimum frequency step that the DCO output frequency can change.

- Nonlinearity: in oscillator frequency tuning, the relationship between oscillator output frequency and the input control value is expected to be as linear as possible. Nonlinearity refers to any deviation of this relationship from an ideal linear function. Nonlinearity in oscillator frequency tuning tends to degrade the performance of the PLL that uses the oscillator, and it is usually minimized as much as possible in design.

- Frequency pushing: frequency pushing is a measure of the sensitivity of the oscillator output frequency to supply voltage fluctuation. It is defined as an increase in oscillator output frequency per unit increase of the supply voltage, and it is typically given in units of MHz/V. A lower frequency pushing is desired to suppress the coupling of power supply noise to the oscillator output.

- Temperature drift: oscillator temperature drift refers to a change in the oscillator output frequency as a result of temperature changes.

- Phase noise: free-running RF oscillator phase noise tends to be the most dominant component of the overall phase noise of an PLL. Particularly at high offset frequencies, the oscillator phase noise shows up without suppression by the loop.

5.2 Oscillator frequency tuning overview

Oscillators are usually designed with a tunable output frequency. The frequency tunability is necessary not only to cover a given frequency range in target applications but also to calibrate PVT variations. In principle, the frequency of an LC oscillator can be tuned through the inductance or capacitance value of the LC tank. In practice, tuning the value of an on-chip inductance has been relatively inconvenient compared to tuning the capacitance. As a result, the frequency tuning of an LC oscillator has been predominantly done through the capacitance tuning, while inductance tuning is occasionally used only when a large tuning range is desired and cannot be fulfilled by capacitance tuning alone.

5.2.1 Frequency tuning in a VCO

In a traditional VCO, the output frequency is tuned using an analog voltage at its input. The voltage-to-frequency relationship can generally be expressed as

$$f_{\text{out}} = f_0 + K_{\text{vco}} \cdot V_{\text{ctrl}}, \tag{5.11}$$

where V_{ctrl} is the input control voltage, K_{vco} is the VCO gain and f_0 is the frequency at $V_{\text{ctrl}} = 0$. For an ideal VCO, the relationship is expected to be linear, as shown in Fig. 5.7, with parameters f_0 and K_{vco} being constant. In real designs, however, they vary together with environmental factors. Particularly, the frequency, f_0, is subject to the impact of temperature drift or supply variation, while the VCO gain, K_{vco}, is itself a function of the applied control voltage. The deviations from the ideal scenario constitute a potential source of performance degradation for the PLL and higher-level systems, and they should usually be minimized with proper design.

With an LC oscillator, the voltage frequency control is usually realized by including a varactor in the LC tank so that the tank capacitance can be tuned by the input voltage. Fig. 5.8 shows examples of LC VCOs with MOS transistors employed as varactors.

Continuous voltage tuning is usually suitable to provide only a relatively small frequency tuning range. This is because the VCO gain, K_{vco}, needs to be low in order to suppress the conversion of the control voltage noise into the oscillator output phase spectrum, and on-chip headroom for the control voltage is nowadays particularly limited with scaled CMOS pro-

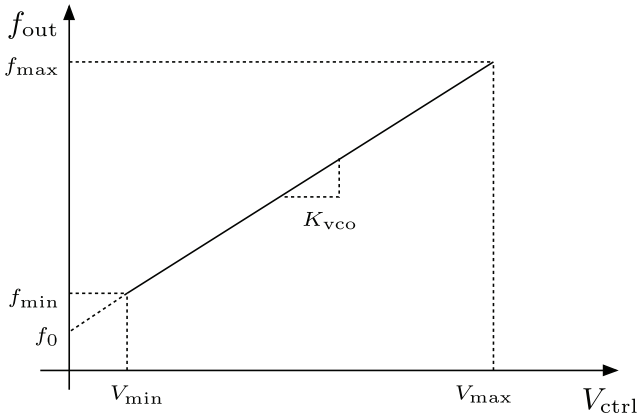


Figure 5.7. Frequency tuning of an ideal VCO.

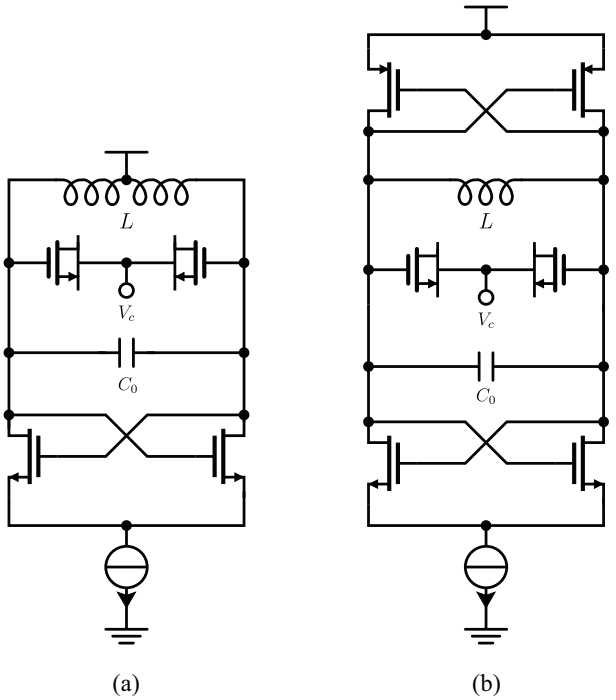


Figure 5.8. Narrow-band LC VCOs: (a) NMOS-core, and (b) CMOS-core.

cesses. As a result, the voltage frequency tuning alone is often not adequate for wideband applications. It has been common with VCO designs to employ digital coarse tuning along with the voltage tuning to extend their frequency range coverage without compromising their noise performance [58, 78–80]. The concept is illustrated in Fig. 5.9, where the gray line and darker lines, respectively, represent voltage frequency tuning without and with digital coarse tuning. The wide-band VCO is functionally equivalent to a group of narrow-band sub-VCOs with staggered but

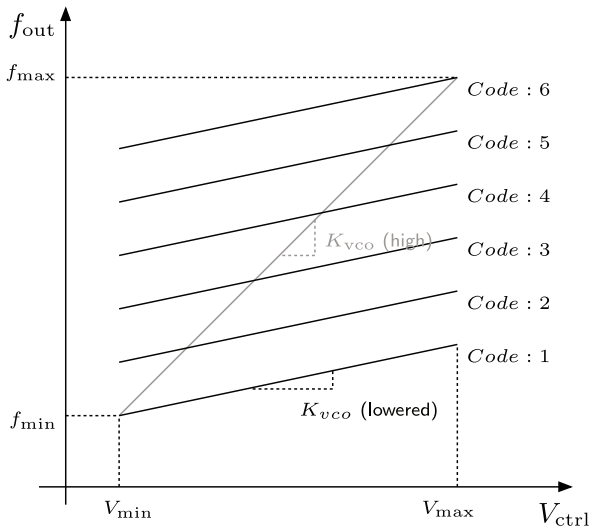


Figure 5.9. Frequency tuning of a VCO with digital sub-band selection.

overlapping frequency tuning ranges, each of which is small but covers a wide overall range when combined. When used in a PLL, the coarse tuning of the VCO must be performed first during the frequency acquisition stage to select an appropriate sub-VCO before the analog voltage tuning can operate. The coarse tuning basically sets the operating point for the subsequent fine analog voltage tuning. A common approach to realizing digital coarse tuning is to use an array of switchable capacitors in the LC tank. An example of a wide-band VCO is shown in Fig. 5.10.

It is probably worth noting that the digital coarse tuning does not change the term "VCO". This is understandable considering the relatively secondary functionality performed by the coarse tuning. During normal operation, the output frequency from a VCO can change continuously without involving quantization errors or requiring a sampling clock for frequency updates, which is in contrast with a DCO.

5.2.2 Frequency tuning in a DCO

With a DCO, analog voltage control is completely avoided by also adopting digital control for fine tuning. The output frequency of a DCO is a function of its digital control words. The digital-to-frequency relationship can be generally expressed as

$$f_{\text{out}} = f_0 + K_{\text{dco}} \cdot D_{\text{ctrl}}, \tag{5.12}$$

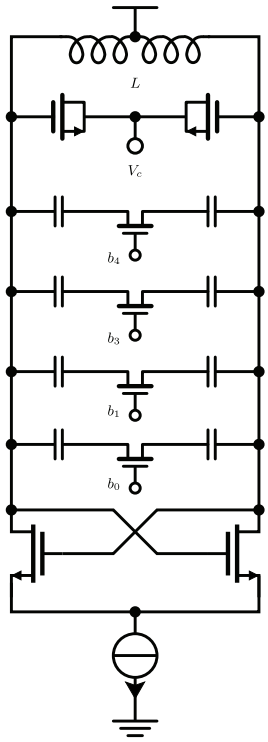


Figure 5.10. A wideband LC VCO.

where D_{ctrl} represents the digital control value and K_{dco} the DCO gain. As with a VCO, the ideal relationship is expected to be linear, with the parameters remaining constant. This ideal relationship is illustrated in Fig. 5.11. However, for realistic designs of DCOs, we have to deal with non-ideal factors similar to those in a VCO. One important consideration is the mismatch among the array of components used to implement digital frequency tuning, which could make the DCO gain, K_{dco}, a strong function of the control value, D_{ctrl}, and thus the relationship strongly nonlinear. The component mismatch limits the feasible frequency tuning range that can be achieved with a single array of components or a single frequency control word. As a result, multiple control words with multiple component arrays for frequency control are usually needed for a DCO to meet the range and resolution requirement, while only one of the control words is active during normal operation and others are for coarse tuning purposes. Realization of digital frequency control with an LC DCO is illustrated in Fig. 5.12.

The evolution from wideband VCOs to DCOs is not as trivial or straightforward a step as it appears, mainly because of the stringent performance

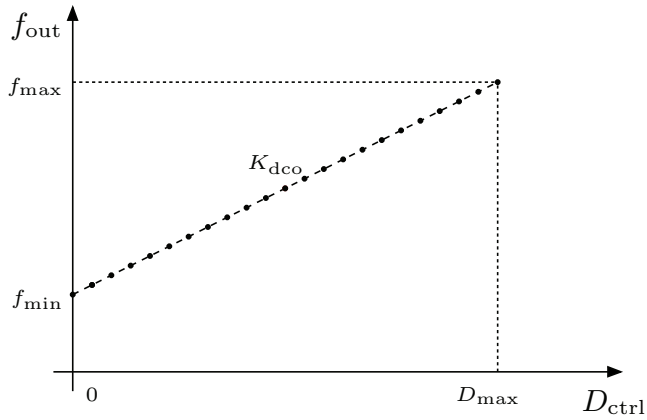


Figure 5.11. Frequency tuning of an ideal DCO.

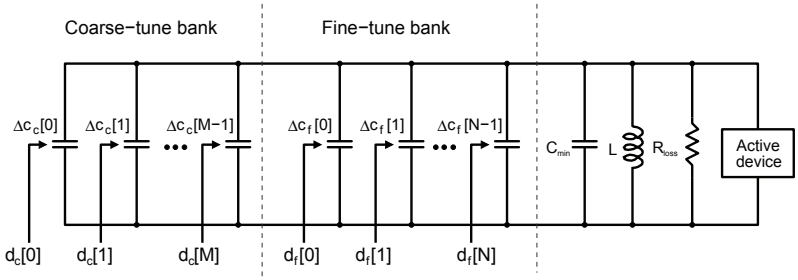


Figure 5.12. An LC oscillator with digital frequency control.

requirements in RF applications. Particularly, frequency synthesizers that employ an RF oscillator in a PLL need to meet strict requirements for phase spectrum purity. On the other hand, frequency quantization and possible glitches associated with DCO frequency control tend to corrupt the PLL phase spectrum and prevent it from meeting the specifications. There are several fundamental limiting factors that make it difficult to address the quantization issues. Only recently have researchers come up with successful DCO-based designs for RF applications [59], thanks to advances of CMOS technology and invention of some novel circuit techniques. These techniques will be discussed in the following sections.

5.3 Performance impact of digital frequency control

Despite the well-known advantages provided by down-scaled CMOS technologies, a DCO has its own plethora of design issues as a result of digital frequency control in the normal operation. A digital approach introduces quantization to both the DCO output frequency and its updating

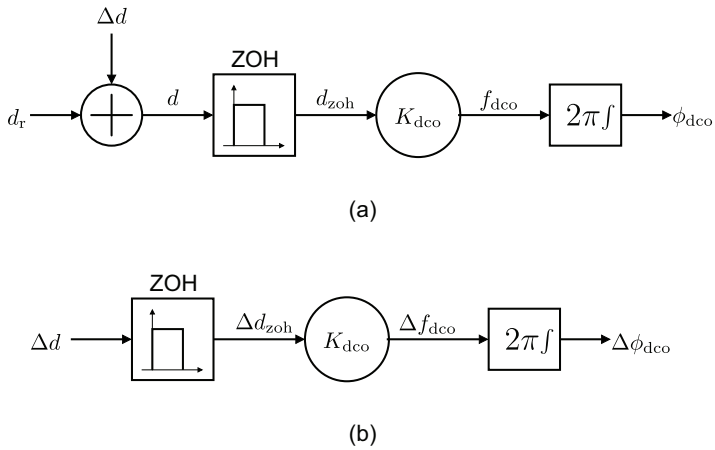


Figure 5.13. A DCO model: (a) ideal DCO and (b) its quantization part.

time, which could potentially degrade the output phase spectrum of a frequency synthesizer. Additional nonidealities in real circuit implementations might also cause further performance degradation. Next, we explore the major impact of digital frequency control. The considerations are similar to those for a DAC, but differ in some respects due to the distinct function of DCOs and the different performance requirements associated with them.

5.3.1 Frequency quantization in an ideal DCO

An ideal DCO can be modeled in a way as depicted in Fig. 5.13. The finite-resolution control code, d, is considered superimposition of a component of quantization error, Δd, on an infinite-resolution control code, d_r. The zero-order hold (ZOH) operation represents the function of a sampling register at the DCO input to keep each sample until the next update. The control signal is translated to the DCO output frequency with a constant gain, K. The output phase is the integral of the frequency over time. The relevant signals in both the time-domain and the frequency domain are illustrated in Fig. 5.14.

The quantization error, Δd, has a peak magnitude of one LSB, denoted as d_{lsb}. In practice, it can be considered uniform distributed noise with a white spectrum. Based on the same linear model, the DCO phase noise contribution at a given offset frequency, Δf, as a result of the quantization

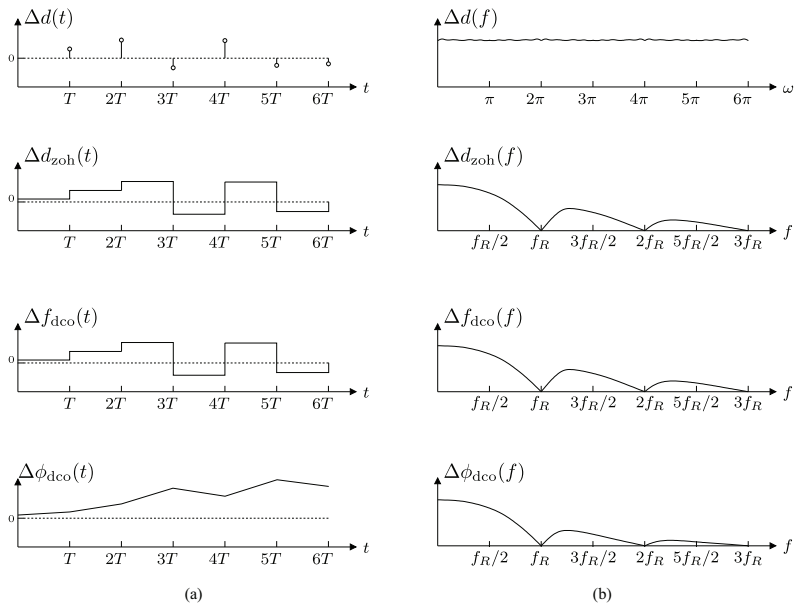


Figure 5.14. Signals in the DCO model: (a) time domain and (b) frequency domain.

effect can be given as follows [59],

$$\mathcal{L}\{\Delta f\} = \frac{1}{12} \cdot \left(\frac{f_{\mathrm{res}}}{\Delta f}\right)^2 \cdot \frac{1}{f_R} \cdot \left(\mathrm{sinc}\frac{\Delta f}{f_R}\right)^2, \tag{5.13}$$

where f_{res} is the quantization frequency step, and f_R is the updating clock rate for the DCO control word. The above equation allows for a few observations:

- A finer frequency resolution or a smaller frequency quantization step, f_{res}, yields lower phase noise contribution. The phase noise contribution decreases by 6 dB for each halving of the quantization step.

  However, a fine frequency resolution however requires small device dimensions that can be below the physical limitation of fabrication technology. In particular, when frequency tuning is achieved with switched capacitors. The minimum frequency step is limited by the minimum switchable capacitance unit that can be reliably fabricated in the process node. Meanwhile, the DCO, even when assisted by coarse tuning, still has to cover a certain frequency range in its fine tuning mode. This frequency range usually needs to be sufficiently larger than one frequency step in the coarse tuning mode to accommodate the temperature drift and supply voltage variations, and in some applications, to allow for direct frequency modulation with the ADPLL. As a result, the relative frequency resolution usually needs to be higher than 10 bits, which makes

it challenging for implementation.

- A higher updating clock rate allows for a lower phase noise contribution. For each doubling of the clock rate, the phase noise contribution can be lowered by 3 dB.

  However, the frequency control signal to the DCO in a PLL is usually propagated from a detected phase error, which is usually updated at the reference frequency. The choice of reference frequency is constrained by multiple system requirements. It involves considering the cost and power efficiency. Moreover, high-frequency crystal oscillators that can be used as frequency references tend to be much more expensive and difficult to fabricate than the more common lower frequency crystal oscillators.

Fig. 5.15 shows an example of quantization phase noise contribution for a realistic updating clock rate, $f_R = 13$ MHz, and DCO frequency quantization step, $f_{\text{res}} = 20$ kHz. It can be seen that the noise contribution level is still too high for a typical wireless application. If the sampling frequency is kept the same, it would be imperative to have a very fine DCO frequency resolution, for example in the sub-kHz region. The fine frequency resolution tends to require small device dimensions, which would be difficult achieve even using current IC fabrication technologies, especially when considering the fact that a DCO still has to cover a certain frequency range in its fine tuning mode. This frequency range usually needs to be sufficiently larger than one frequency step in the coarse tuning mode to accommodate the possible frequency drift due to variations in the supply voltage and operating temperature, and in some applications, to allow for direct frequency modulation with the ADPLL. As a result, the relative frequency resolution usually needs to be higher than 10 bits, which makes it challenging to implement the oscillator core, especially in terms of the physical layout.

It should be noted that the linear additive model is not always practically valid to account for the DCO quantization effect. Likewise, the result given by Eq. (5.13) and other analysis based on the linear model only provide an approximation [55]. With a large frequency quantization step and low sampling frequency, the linear model would break down and it would be impossible to ignore the effect of the inherent nonlinearity of

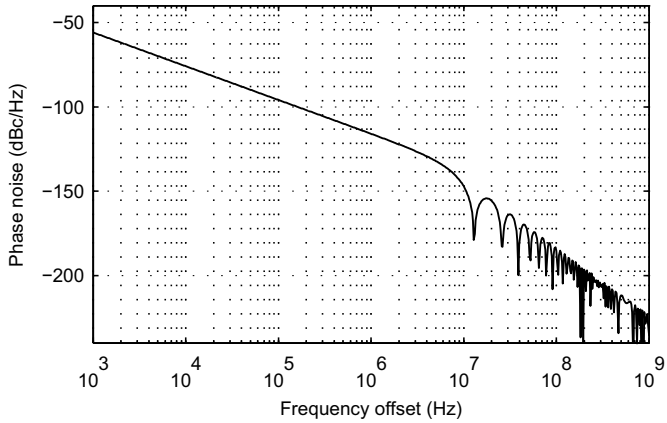


Figure 5.15. DCO phase noise due to frequency quantization ($f_R = 13$ MHz and $f_{\text{res}} = 20$ kHz).

DCO quantization. The nonlinearity of DCO quantization tends to further degrade the phase noise performance. However, the general conclusion that can be drawn when using Eq. (5.13) is typically valid for a close approximation.

5.3.2 DCO non-idealities

The DCO discussed above is an ideal case with perfect linearity. In practice, the unit components used to realize digital frequency control are subject to random and systematic mismatches in IC fabrication and circuit layout. Such mismatches introduces nonlinearity into the digital-to-frequency conversion process. In other words, the DCO gain, K_{dco}, as in Eq. (5.12), is not a constant; instead, it varies with the control code. The nonlinearity of the DCO could be a performance-limiting source of errors for the fine tuning in instances where spectrum purity is typically subject to stringent requirements. It tends to introduce limit cycles to a PLL operation, and as a result, it generates spurious tones in the output phase spectrum.

Component and wiring mismatches can also lead to timing skew among individual bits of control word. As a result, glitches can arise with each update of the control word, particularly when the control word for the DCO core is a binary code.

To improve linearity and to avoid large switching glitches, well-considered layout techniques are often necessary in order to achieve optimal matching of the relevant components for frequency control. Meanwhile, the

DCO control word is usually converted into thermometer code to ensure monotonicity and to avoid large switching glitches. Dynamic element matching (DEM) measures are often employed to break spurious tones that might otherwise arise as a result of the nonlinearity.

5.4 Digital techniques for DCO

5.4.1 DCO frequency dithering

DCO frequency dithering is a technique to improve the effective frequency resolution of the DCO by upsampling and typically also noise shaping. It upsamples a relatively high-resolution input control word from the relatively low frequency domain, quantizes it to match the relatively coarse DCO frequency step and switches the DCO output frequency at the upsampling clock rate to create a time-averaged frequency value that corresponds to the input control word value. It makes possible a high-resolution control word in the reference clock domain without requiring a correspondingly fine DCO frequency step.

The basic idea is illustrated in Fig. 5.16. The DCO output frequency alternates between two adjacent discrete values of f_n and $f_n + \Delta f_{\mathrm{lsb}}$ several times during a reference frequency cycle, rather than assuming a constant value of either f_n or $f_n + \Delta f_{\mathrm{lsb}}$. The time-averaged value of the DCO output frequency is $f_n + \Delta f_{\mathrm{lsb}}/4$, which means that the effective resolution becomes $f_{\mathrm{lsb}}/4$, i.e. one-fourth of the quantization step. This basic DCO dithering can be realized by applying the control word to the input of a simple modulo accumulator, as shown in Fig. 5.17, with its 1-bit carry-out to control a 1-LSB DCO frequency step. The simple modulo accumulator essentially forms a 1st-order digital sigma-delta modulator (DSDM). Based on the linear model, its transfer function as can be written as

$$Y(z) = z^{-1}X(z) + \left(1 - z^{-1}\right) z^{-1}E(z), \tag{5.14}$$

where $X(z)$, $Y(z)$ and $E(z)$ are, respectively, the z-transform of the input, output and quantization error. It should be noted that there is an additional unit delay, z^{-1}, at the output, which is usually needed in order to synchronize of the digital signal update before the signal goes to the oscillator core for frequency control. For a M-bit DSDM, the achievable

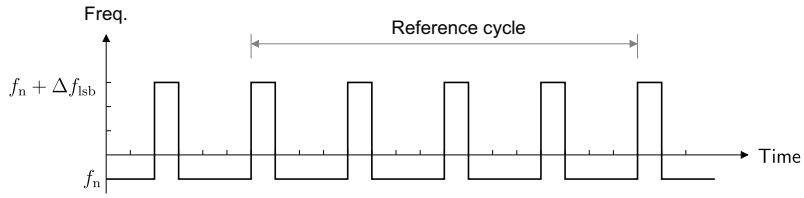


Figure 5.16. An example of basic DCO frequency dithering.

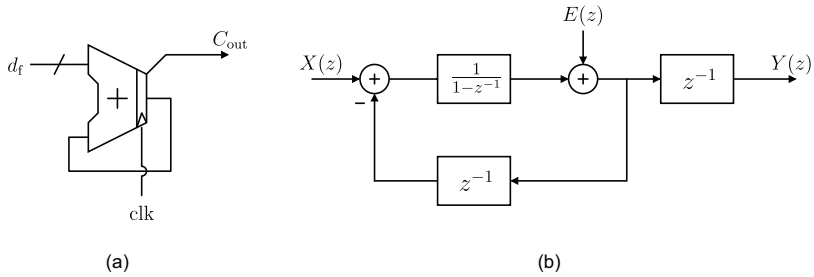


Figure 5.17. Accumulator for the basic dithering: (a) circuit and (b) linear model.

effective frequency resolution, f_{res}, can be 2^{-M} of the basic DCO frequency step, f_{lsb}. However, as indicated in Fig. 5.17, the clock rate for the DCO dithering typically needs to be much higher than that of the reference clock in order to make the dithering effective. This can also be seen from the DSDM transfer function, where the magnitude of the output is equal to the input plus a high-passed component of the quantization error. Basically, the quantization error becomes insignificant from the perspective of the reference clock domain only if the dithering clock rate is much higher. As pointed out by Staszewski et al [59], the dithering speed should be higher than the reference clock rate multiplied by the factor of resolution improvement. In other words, to achieve an M-bit fractional frequency resolution, the clock rate needs to be at least 2^{-M} times higher than the reference frequency. This becomes clearer when the phase noise contribution of the DSDM itself is taken into account. The clock can be generated from the DCO output with an appropriate frequency division ratio.

A well-known issue pertaining to the simple dithering or the first-order DSDM is that it is highly periodic and thus prone to generate spurious tones at the DCO output. To allow for spurious-free output, a higher-order DSDM is often necessary for the realization of DCO dithering. The multi-stage noise shaping (MASH) DSDM architecture is usually preferred for DCO dithering due to its simplicity and cost-effectiveness. Fig. 5.18 shows hardware realization for the second-order MASH (MASH 1-1) and the third-order MASH (MASH 1-1-1) DSDMs, which have been the most com-

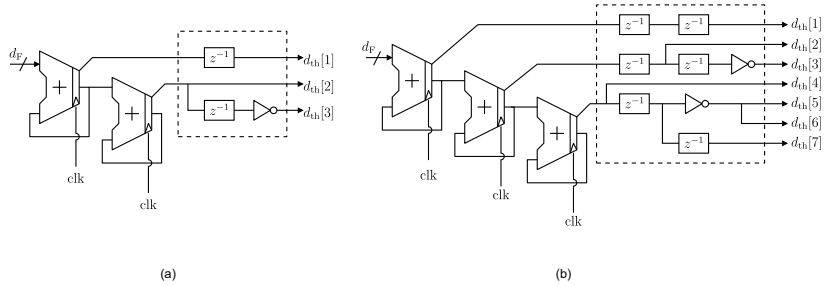


Figure 5.18. MASH SDM for DCO dithering: (a) 2nd-order and (b) 3rd-order.

monly used DCO dithering DSDMs. It should be noted that the DC level of the SDM output is usually shifted up with respect to the input so as to avoid possible negative values that cannot be physically realized. Any DC-level shift introduced to the output coding has just some negligible impact on the initial DCO frequency, which is of no significance in practice. In addition, the output usually takes the form of a thermometer code instead of a binary code, not only to minimize glitches and mismatch effects, but also to obviate the need for further arithmetic addition operations in order to combine the output bits. To ensure spurious-free output, different dithering techniques can be employed for the SDM [81–83].

While frequency dithering refines the effective DCO resolution, it also makes its own contribution to the DCO output phase noise, which can be approximated as [54]

$$\mathcal{L}\{\Delta f\} = \frac{1}{12} \cdot \left(\frac{\Delta f_{\text{res}}}{\Delta f}\right)^2 \cdot \frac{1}{f_{\text{dith}}} \cdot \left(2 \sin \frac{\pi \Delta f}{f_{\text{dith}}}\right)^{2n}, \tag{5.15}$$

where n is the order of SDM. The overall quantization noise effect is the combination of both the SDM phase noise contribution and that of Eq. (5.15) with a refined effective frequency resolution. Examples of phase noise contribution with SDM DCO dithering are given in Fig. 5.19 and Fig. 5.20.

5.4.2 DCO linearization with DEM

Dynamic element matching (DEM) is widely used to linearize D/A converters [84–87]. Similarly, the DEM can also be employed to linearize a DCO by averaging the errors out through the randomized selection of unit components. The choice of a specific DEM method often involves making a trade-off between performance and complexity. Among the dif-

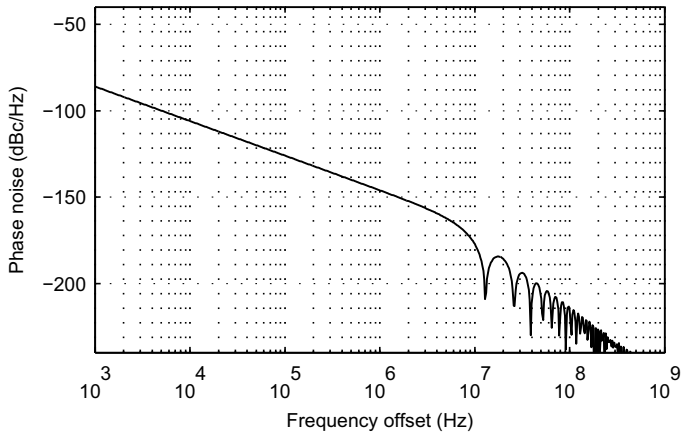


Figure 5.19. DCO phase noise due to frequency quantization with 5 fractional bits ($f_R = 13$ MHz and $f_{\text{res}} = 20$ kHz).

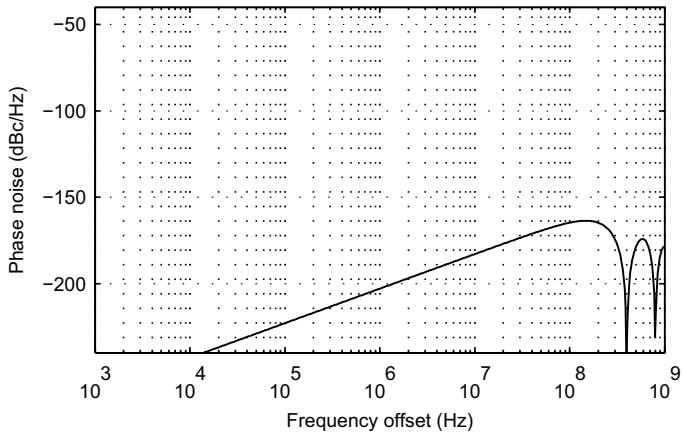


Figure 5.20. DCO phase noise due to with 2nd-ord SDM frequency dithering ($f_{\text{dith}} = 800$MHz and $f_{\text{res}} = 20$ kHz).

ferent DEM methods, the data-weighted averaging (DWA) with a cyclic shift of element selection is one that features relatively low complexity and high efficiency [88]. A typical DCO for RF applications usually requires no less than a 10-bit resolution, for which a DWA with full-scale cyclic shift can still be too costly for hardware realization. For further complexity reduction, a partial DWA schemes with segmented implementation are often preferred. Fig. 5.21 illustrates the element selection and circuit realization of a partial DWA scheme for a DCO with a 6-bit binary input. A variant of such DWA schemes was also adopted in previous studies [3] for the integer part of a binary fine frequency tuning word. When the shift in the element selection is limited within the last matrix row, the size of the barrel shifter is greatly reduced. The use of partial DWA can be justified considering the slow variation in the DCO frequency and thus

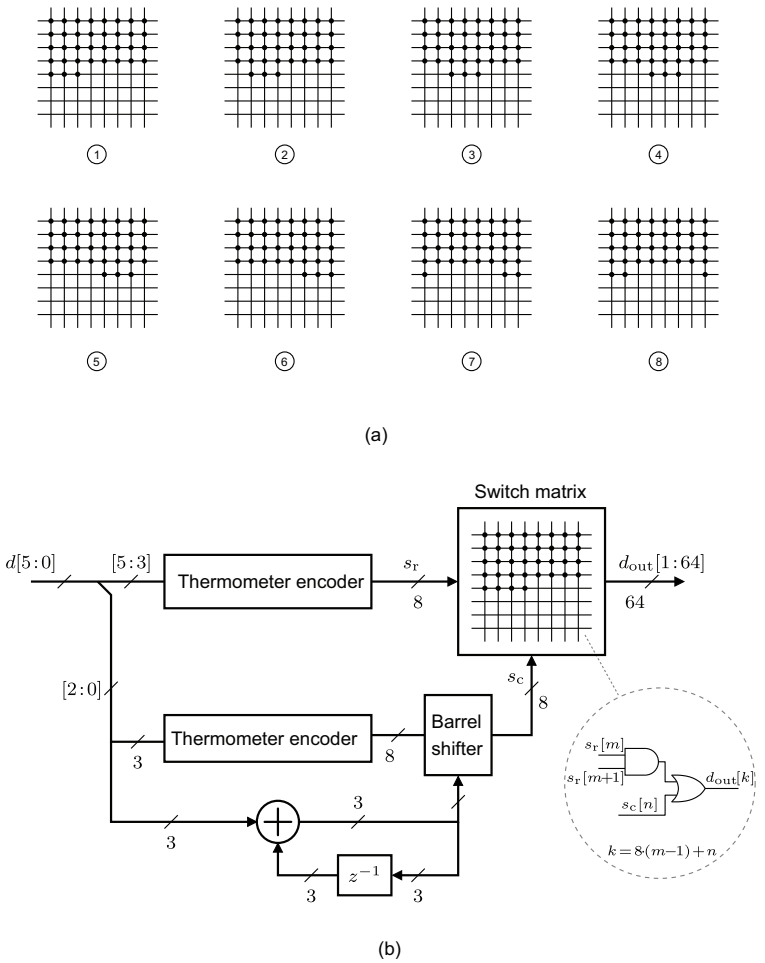


Figure 5.21. A partial DWA for DCO: (a) operation and (b) circuit realization.

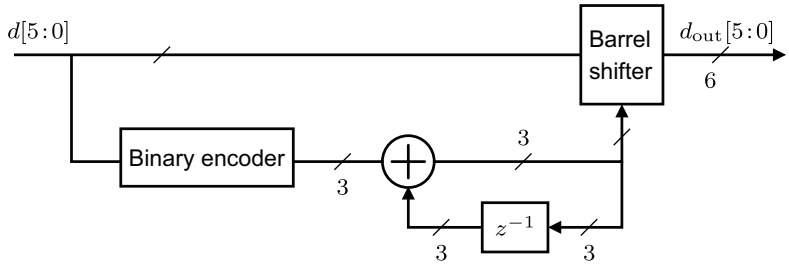


Figure 5.22. A DWA with thermometer-code input.

the digital control signal during a PLL steady-state operation where the output spectrum matters. For a SDM output with a small word length and thermometer code, a local full-scale DWA can be used, as shown in Fig. 5.22.

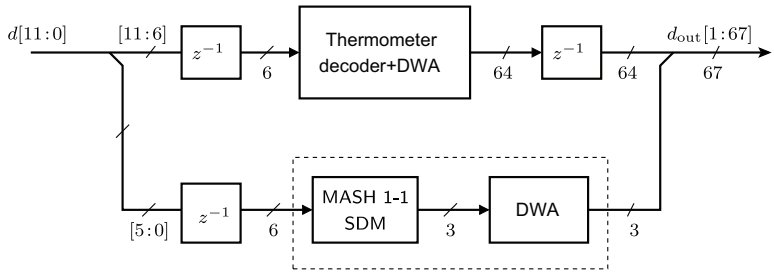


Figure 5.23. An example of DCO interface logic

5.4.3 Digital interface logic

The above techniques for boosting DCO dynamic performance can be implemented as interface logic. An example of such implementation is given in Fig. 5.23. Basically, the input frequency tuning word can be viewed a a fixed-point format. Its fractional part, after sampling by the reference clock, is fed into a digital SDM to for frequency dithering. The SDM output is a thermometer code, which goes through the DEM block to randomize its mapping to unit elements in the DCO. The SDM and the following DWA blocks operate in a high-rate clock domain. Such a clock can be derived from the DCO output through frequency division. The integer part of the frequency tuning word is thermometer coded with partial DWA, which is then sampled by the reference frequency clock at the output.

5.5 Summary and discussion

The difference between a VCO and a DCO resides mainly in their fine frequency tuning approaches. The use of digital frequency control in a DCO, as opposed to analog voltage control in a VCO, requires new design considerations. Frequency quantization with digital control constitutes a source of phase noise. In a typical RF application, the phase noise contribution of frequency quantization needs to be well below the intrinsic phase noise of the oscillator core. This require small quantization steps, which could be difficult to realize together with the frequency range necessary to cover the target frequency band with the presence of PVT variations. Frequency dithering with a $\Sigma\Delta$-modulator could effectively create fractional frequency resolution, their by making the DCO core design easier. In addition, component mismatch could be another limiting source of errors. A common approach to linearizing the DCO is the use of DEM

schemes. The choice of DEM schemes is mainly a trade-off between performance and complexity. It should be noted that a DEM basically breaks the power of possible spurs into noises.

6. Phase-to-digital converter

The feedback path of the ADPLL serves as a frequency-to-digital converter (FDC) that quantizes the DCO output frequency. The core of such an FDC is a phase-to-digital converter (PDC), which converts the DCO output phase in the continuous-time domain into digital form in the discrete-time domain. The ideal function of a PDC is described by Eq. (3.4), while it is generally characterized by different non-ideal factors in a real implementation. The PDC performance is often crucial to the overall performance of the ADPLL and its design constitutes one of the most demanding challenges. This chapter examines the key aspects of PDC design and operation both specific to the ADPLL architecture and in general terms.

6.1 Common approaches

The DCO output can be considered to be a sinusoidal function of its increasing phase over time, as given by

$$V(t) = V_0 \sin\left[\phi_v(t)\right], \tag{6.1}$$

where V_0 is its peak amplitude and $\phi_v(t)$ is the phase at time, t. For the digitization of $\phi_v(t)$ by a PDC, the zero-crossing points of the sinusoidal waveform play a special role. According to Eq. (6.1), they are the points where $\phi_v(t) = n\pi$ with n an integer index of the points. In a circuit implementation, the DCO output amplitude is usually clipped by the subsequent buffers or amplifiers before the signal is subject to the phase digitization. The resulting waveform often features sharp rising or falling edges. It allows for digital circuit devices, such as flip-flops, which are based on clock edge-transitions for operation, to be employed in the circuit for accurate detection of the zero-crossing points.

6.1.1 Counting-based approach

A counting-based PDC simply counts the zero-crossing points and samples the counting result at the reference clock. Typically, only the rising or falling edges but not both are counted for practical considerations. As a result, the output of a PDC based on this approach can be given as

$$\Phi_{v,I}[k] = \frac{\phi_v(t_s[k])}{2\pi} + E_{q,I}[k], \tag{6.2}$$

where k is the sample index, $t_s[k]$ is the kth sampling time and $E_{q,I}[k]$ is the quantization error. Since only the whole cycles are counted and the entire fractional part is quantized off, the quantization error can be given by

$$E_{q,I}[k] = -\left(\frac{\phi_v(t_s[k])}{2\pi} \bmod 1\right). \tag{6.3}$$

The modulo operator indicates the quantization error corresponds to the fractional fraction cycle of the phase. Fig. 6.1 (a) illustrates the operation principle of a counting-based PDC. In a practical implementation, the counting is typically performed by a modulo counter or accumulator clocked by the DCO signal, while the output sampled by a register clocked at a reference frequency. The modulo counter wraps around whenever its maximum value is reached. The output of the PDC based on an N-bit counter can be written as

$$\hat{\Phi}_{v,I}[k] = \left(\frac{\phi_v(t_s[k])}{2\pi} + E_{q,I}[k]\right) \bmod 2^N. \tag{6.4}$$

With proper choice of N, the modulo operation in Eq. 6.4 can be undone in the subsequent logic in the frequency error detection of an ADPLL and thus it constitutes no consequence, as we elaborated in a published study [13].

Since the resulting PDC accumulates the variable phase of the DCO, it is often called variable phase accumulator (VPA) in the existing literature [5, 49]. We will also refer to it as an integer PDC since it counts the integer number of cycles and outputs an integer number.

6.1.2 Time-measurement approach

The DCO frequency within one cycle of the DCO output signal can be considered a constant. The phase progression corresponding to the fractional

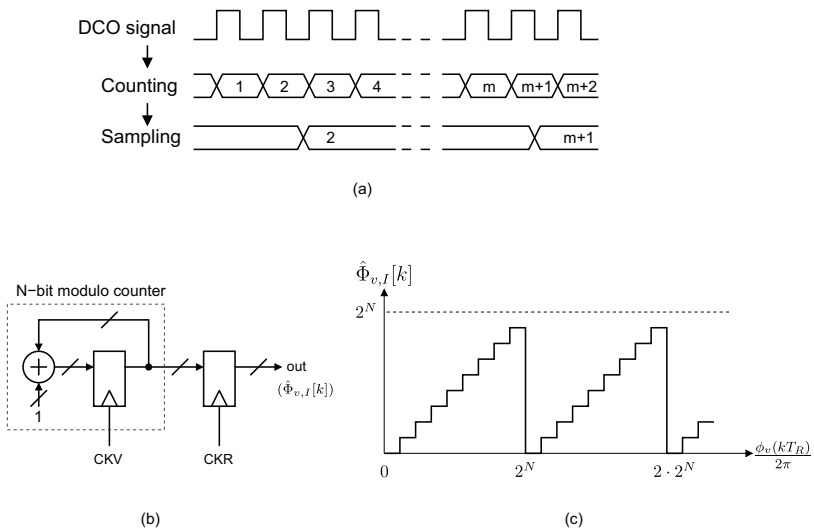


Figure 6.1. Principle of a counting-based PDC: (a) operation principle, (b) practical hardware realization and (c) output relationship to the DCO phase.

cycle quantized off by the counting-based PDC, which is the term in the bracket in Eq. (6.3), can be calculated as

$$\frac{\phi_v(t_s[k])}{2\pi} \bmod 1 = \tilde{f}_v[k] \cdot \Delta t_r[k] = \frac{\Delta t_r[k]}{\tilde{T}_v[k]}. \tag{6.5}$$

The $\Delta t_r[k]$ is the time interval from the preceding zero-crossing point to the sampling point, while $\tilde{f}_v[k]$ and $\tilde{T}_v[k]$ are, respectively, the DCO frequency and period at the moment of measurement. The observation leads to the realization of a PDC based on time-to-digital converter (TDC). It measures relevant time intervals and calculate the phase progression. In other words, the PDC output is given by

$$\Phi_{v,F}[k] = \left(\frac{\phi_v(t_s[k])}{2\pi} + E_{q,F}[k] \right) \bmod 1, \tag{6.6}$$

where $E_{q,F}[k]$ represents a relatively small quantization error of the PDC. As demonstrated also in our earlier work [13], the effect of the modulo operation in above equation can be canceled out in the subsequent logic and the PDC range of one cycle is sufficient if the following condition is satisfied: the difference between the DCO frequency and the target frequency is smaller than half of the reference frequency by a margin corresponding to the magnitude of quantization errors. Since the condition is usually satisfied after initial frequency settling of the ADPLL, our work essentially proved that the TDC-based approach alone is adequate for the ADPLL operation after frequency acquisition.

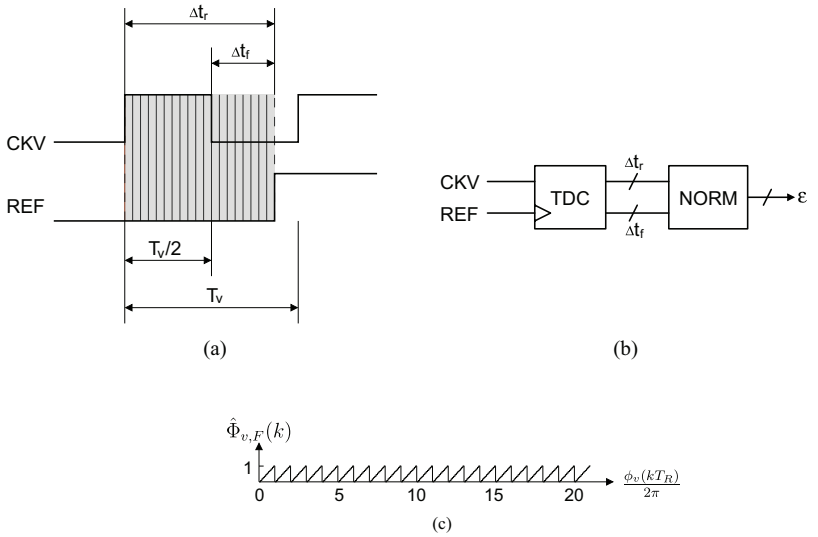


Figure 6.2. Time-measurement-based PDC approach: (a) operation principle, (b) a possible realization and (c) output relationship to DCO phase.

Fig. 6.2 shows the principle of a time-measurement approach, along with its basic hardware realization and its output as a function of the DCO phase. The time measurement is generally performed by a time-to-digital converter, and measured time intervals are used by a normalization logic block to derive the necessary output in the digital domain. As expected, the output is a fractional number.

6.1.3 Combined approach

The above two approaches appear to be complementary to each other since one covers the whole cycles and the other covers the fractional part. Another common approach has thus been based on the combination of the above two approaches. In an ideal case, the overall output of the PDC, as the sum of Eq. (6.4) and Eq. (6.6), can be written as:

$$\Phi_{v,C}[k] = \left(\frac{\phi_v(t_s[k])}{2\pi} + E_{q,F}[k] \right) \mod 2^N, \tag{6.7}$$

where $E_{q,F}[k]$ is the small quantization error from the time-measurement PDC. As can observed from Eq. (6.7), the resulting PDC achieves a fine resolution determined by the time-measurement approach, while it covers the same range as the counting-based approach. A combined PDC and its ideal output is shown in Fig. 6.3 (a) and (b).

The ideal case is based on the following assumptions: first, the sampling

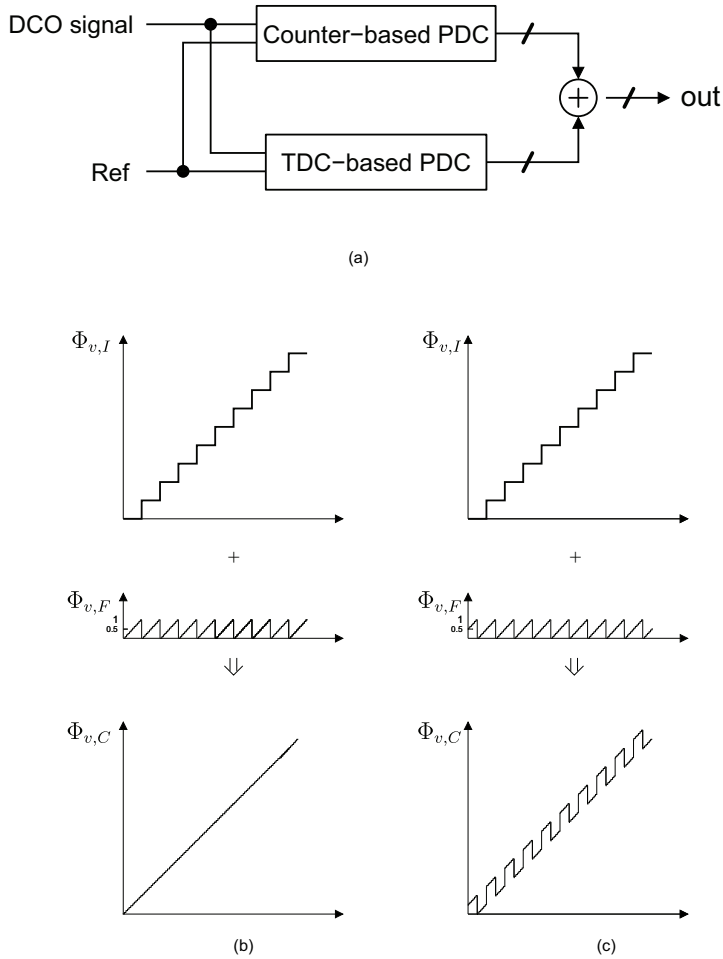


Figure 6.3. A PDC based on a combined approach: (a) hardware realization, (b) output in an ideal case, and (c) output in a realistic case.

time, represented by $t_s[k]$, is exactly the same in the two PDC parts; second, the quantization error, $E_{q,F}[k]$ in Eq. (6.6), does not cause an overflow or wrapping-around related the modulo operation. However, neither of the assumption is realistic. In a real case, as shown in Fig. 6.3 (c), the transfer function of the PDC is distorted due to the violation of the above assumptions. In the time domain, the distortion manifests itself as glitches with a large magnitude of one. If not corrected, they essentially null all the benefits from the fine resolution with the TDC-based approach. The known remedy for this is to use a glitch correction circuit at a later stage following the frequency error detection in the loop [12, 89–91]. It relies on the fact that the frequency error is much smaller than half of the reference frequency when the ADPLL is in locked state. Clearly the glitch correction does not work during the coarse tuning stage, where the

glitches have to be tolerated.

As opposed to the combined approach, we proposed separate use of the integer PDC and fractional PDC and the TDC-based approach respectively in coarse and fine tuning modes to avoid unnecessary complication with the combination [13].

6.2 Time-to-digital converter

The time-to-digital converter (TDC) is the most critical block in a PDC. Its performance to a large extent determines the PDC performance and thus the overall performance of an ADPLL. Most of the PDC design challenges are associated with the TDC design. In addition, the TDC can also be used in laser ranging, nuclear and particle physics, 3-D imaging, and so on. As a result, it has been the subject of intensive research efforts in recent years. This section reviews TDC common architectures and design considerations.

6.2.1 General considerations

6.2.1.1 Basic principle

A TDC measures the time difference or time interval between every two events and presents the results in digital form. As shown in Fig. 6.4, the relevant events are defined by the leading and trailing edges, which are typically provided in separate signal lines. Ideally, the transfer function of a TDC is in the form of a uniform quantizer that maps the continuous-valued input time interval to a discrete digital output. The relationship can be given as

$$D_{\text{out}} = \frac{T_{\text{in}}}{T_{\text{lsb}}} + E_q, \tag{6.8}$$

where T_{lsb} is the range of input change corresponding to one LSB in the TDC output and E_q is the quantization error.

A TDC is often compared to an analog-to-digital converter (ADC). They are similar in that they both convert a signal into the digital domain. Meanwhile, there are fundamental differences between them as dictated by the different natures of their respective input signals. These differences include:

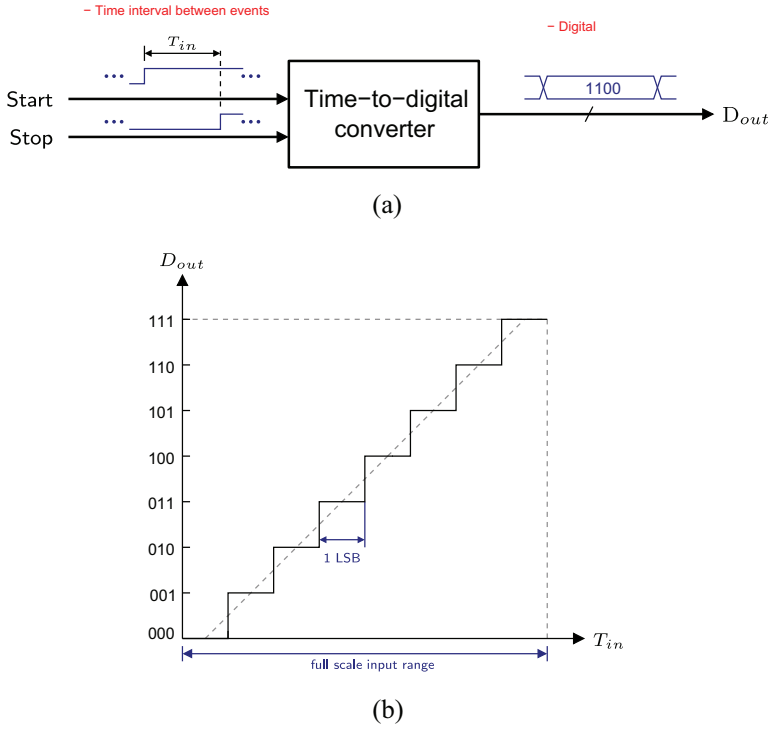


Figure 6.4. TDC principle: (a) basic diagram and (b) transfer function.

- Storage of time or time intervals is practically impossible before its conversion into another form. This is in contrast with relative ease of storing an analog voltage. In this respect, the time interval as a signal is more like a current than a voltage.

- Events to a TDC are by nature discrete in the time domain, unlike an analog voltage signal, which can be distributed continuously over time.

- Measurement of time difference involves two events spaced in time. As a result, the acquisition of one time interval as input to a TDC takes a corresponding amount of time. This is in contrast with an ADC, where a voltage value can in principle be acquired instantly when it is present.

These difference and others are reflected in how a TDC is realized and characterized.

6.2.1.2 Performance parameters

The characterization of a TDC is comparable to that of an ADC. While ADC characterization is relatively well standardized, no standardization exists for TDC characterization. Most of the TDC performance parameters are based on those used for the ADC characterization, though some of the parameters have been reinterpreted. Commonly, these parameters are classified as static and dynamic parameters. Static parameters are those that can be tested with a repeated input value at a sufficiently low speed without active peripheral circuits. The most relevant static parameters in TDC characterization include time resolution, the conversion range and nonlinearity errors (DNL and INL). Dynamic parameters can be further grouped into time-domain dynamic parameters and frequency-domain dynamic parameters. Time-domain dynamic parameters include conversion time, latency and dead time. Frequency-domain dynamic parameters for an ADC include the signal-to-noise ratio (SNR), the effective number of bits (ENOB) and signal-to-distortion-and-noise (SNDR). However, there are some difficulties in applying these parameters to a TDC. A more comprehensive discussion of the relevance of ADC parameters and their test procedures in TDC characterization can be found in a study by Henzler [92].

The most fundamental parameter for a TDC is probably time resolution, which refers to the TDC's LSB time step, T_{lsb}. The time resolution basically determines the level of TDC quantization errors. Time resolution for a TDC is essentially what voltage resolution is to an ADC. The range or measurement range is another fundamental TDC parameter, which is the maximum time interval measurable with a given TDC implementation. A related term is resolution. Resolution is often used informally as shorthand for time resolution. However, resolution by definition refers to the bit number that corresponds to the total number of quantization levels over the full scale measurement range.

It is necessary to compare the above TDC parameters to their counterparts in order to clarify their significance. It should be noted that the ADC's measurement range is typically determined by its voltage reference and limited by the supply voltage level. Once the voltage reference has been chosen, the ADC voltage resolution is basically the voltage range divided by the number of realized quantization levels. However, both the measurement range and the voltage resolution can effectively be altered

with an amplifier or attenuator placed in front of the ADC for the input voltage signal. As a result, they are not of critical importance. What is significant is the number of quantization levels for an ADC, which is given as the ADC resolution in bit numbers, e.g. a 5-bit ADC has $2^5 = 32$ quantization levels. The situation is different for a TDC, since amplifying the time interval before an TDC, especially with the necessary degree of accuracy, is much more challenging than amplifying the voltage. Therefore, the time resolution of a TDC is much more critical than the voltage resolution of an ADC. Given the information about time resolution, the range and resolution can basically be derived from each other and thus only one of them can be provided as a separate parameter. It is also important to note that there is not always a time interval reference like the voltage reference in an ADC that clearly determines the measurement range of a TDC. In the case of an ADPLL, such a reference time interval can be considered as one cycle of the RF feedback signal, one that is present in the input signal. As a result, the time interval reference varies with the frequency of the RF signal to the TDC and is not directly related to the TDC measurement range.

The ideal TDC has uniform quantization steps, with the midpoint (or endpoint) of each step lying on a straight line. Any deviation of the real TDC transfer function from this ideal linearity is usually characterized by differential nonlinearity (DNL) and integral nonlinearity (INL). As shown Fig. 6.5, the definitions of DNL and INL for a TDC are similar to the definitions for an ADC. The DNL is the deviation of an actual step width from the ideal one (1 LSB). The INL is the cumulative DNL. It measures the deviation of the entire transfer function from the ideal one. Both the DNL and INL are referenced in relation to the LSB. It should be noted that any deviation of the actual TDC gain from an ideal gain is referred to as TDC gain error, which is a measure of TDC linear imperfection. In an ADPLL with a periodic TDC input time interval, the TDC linear imperfection, such as the gain error, can be translated into nonlinear distortions, with the output effectively unwrapped in the digital domain.

The time-domain dynamic parameters can be used to characterize the timing behavior of a TDC. The conversion time, T_{conv}, is defined as the time between the start event and the availability of the digital output. The latency, $T_{latency}$, refers to the delay between the stop event and the availability of the digital output. The dead time, T_{dead}, refers to the time

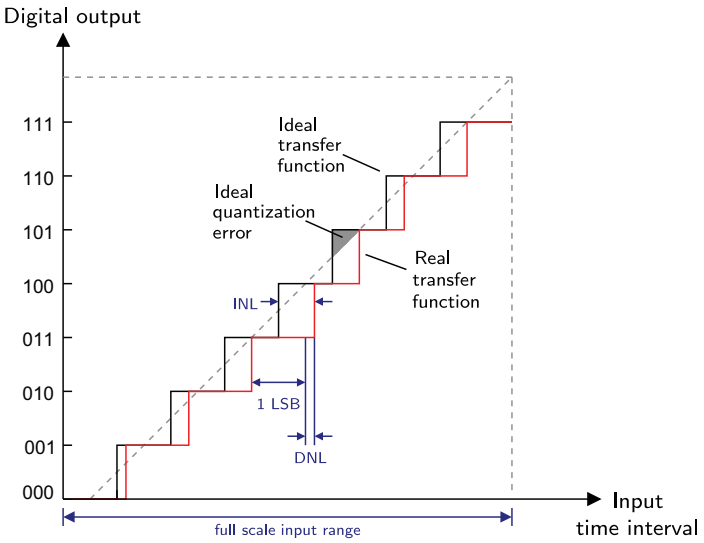


Figure 6.5. Static characteristic of an ideal TDC.

it takes a TDC to be ready for a new conversion after one conversion is finished. The maximum conversion rate of a TDC is limited by $T_{conv} + T_{dead}$.

Application of classical frequency-domain parameters of an ADC to a TDC is less straightforward. It requires an accurate phase modulation to generate a sinusoidal sequence of time intervals with sufficient accuracy in the TDC simulation or measurement. To simplify the test procedure, the concept of a single-shot experiment is proposed for estimating TDC dynamic performance by Henzler in his study [92]. However, the approach requires significant effort in post-processing and does not cover some essential dynamic impacts of the input signals. Given the information about ENOB, the figure of merits for the TDC can be evaluated respectively as,

$$FOM_P = \frac{P}{f_s \cdot 2^{ENOB}}, \tag{6.9}$$

and

$$FOM_A = \frac{A}{f_s \cdot 2^{ENOB}}, \tag{6.10}$$

with P and A being the power consumption and silicon area of the TDC and f_s being the sampling frequency.

6.2.2 Common architectures

In the following section, we are going to examine common TDC architectures resulting from extensive research efforts. Their strengths and weak-

nesses will be highlighted and their suitability for an ADPLL application will be discussed. Among all these architectures, the inverter delay-line-based one has been selected as the base architecture in the author's work because of its simplicity and suitability for power gating [12].

6.2.2.1 Time-to-analog-based architectures

One general approach of time-to-digital conversion is to convert the time interval into a voltage with a time-to-analog converter (TAC) and then quantize the voltage with a conventional ADC. The basic principle is illustrated in Fig. 6.6. For time-to-analog conversion, a pulse generator produces a constant current pulse with a width corresponding to the input time interval, and a subsequent integrator then integrates the current pulse to form a voltage. Despite its conceptual simplicity, the TDC has relatively high complexity when it comes to implementation. It does not lend itself well to the simultaneous conversion of more than one time interval as needed in an ADPLL. Most importantly, it is by nature analog intensive and thus does not have good scalability with CMOS technologies. In a digital-centric design, such as an ADPLL, it is necessary to have a digital-intensive TDC, which is commonly based on digital delay lines.

6.2.2.2 Delay-line-based architectures

If the timing event can be repeated with a certain amount of delay for the time interval that needs to be measured, the number of repetitions is a digital representation of the time interval. With the timing event represented by a rising or falling signal transition edge, its repetition can be readily performed with one or more delay lines. This is the basic idea behind delay-line-based TDC architectures. Since the repetition of timing events with a delay line does not rely on a detailed level of analog voltage, delay-line-based architectures tend to be highly digital or digital-intensive.

6.2.2.2.1 Buffer delay-line TDCs Fig. 6.7 shows a simple buffer delay-line-based TDC. The transition edge of the start signal propagates through a chain of delay cells to create repeated versions with a delay and changes the state of the delay cells that it passes by. Upon arrival of the stop signal, the sampling flip-flops clocked by it take a snapshot of the state of the delay line. A delay cell in the delay line would yield a high value if it has been traversed by the delayed start signal; otherwise it would yield

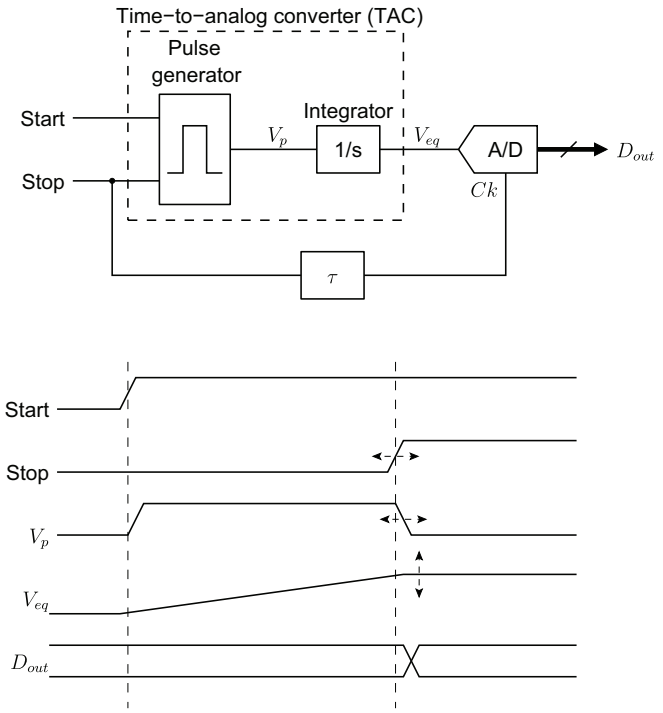


Figure 6.6. A TDC based on a TAC and an ADC.

a low value. The number of high-state cells or the position of the last high-state cell in the thermometer code is thus a measure of how far the start signal could propagate during the input time interval. The result is a thermometer-code representation of the relevant time interval. Suppose $\tau_{\rm lsb}$ is the delay of a unit cell, i.e. a buffer. The propagation speed of the signal along the delay line in terms of taps per second would be $1/\tau_{\rm lsb}$. The distance that the event can travel over a given amount of time is simply the product of time and the traveling speed. As a result, for the time interval, ΔT, as defined by the input timing events, the number of delay cells that would yield a high value because of the passage of the timing event is simply

$$M = \lfloor v \cdot \Delta T \rfloor = \lfloor \frac{\Delta T}{\tau_{\rm lsb}} \rfloor = \frac{\Delta T}{\tau_{\rm lsb}} \bmod 1. \tag{6.11}$$

The floor operation, or the modulo-1 operation in the above equation, is simply the result that the buffer output is high only when the event has passed by and it is low otherwise. The thermometer code can be binary encoded with digital logic. Alternatively, the above relationship can be

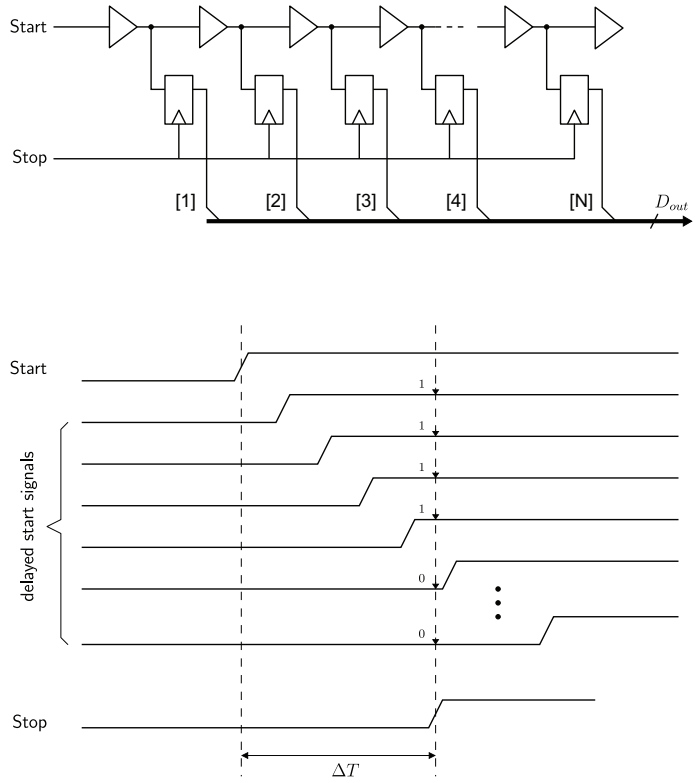


Figure 6.7. A buffer delay-line-based TDC architecture

rewritten to express the time interval, ΔT, as

$$\Delta T = M \cdot \tau_{\mathrm{lsb}} + e_{\mathrm{q}}, \tag{6.12}$$

with e_{q} being the input-referred quantization error.

6.2.2.2.2 Inverter delay-line TDCs Using the buffer as the unit cell, the TDC's time resolution is limited to one buffer delay or two inverter delays. A finer resolution of one inverter delay instead of two can be achieved by using the inverter as the unit delay cell. This usually entails a differential or pseudo-differential structure to minimize the nonlinear impact of the uneven delay by an inverter for the rising and falling edges. Fig. 6.8 shows the principle of such an inverter delay-line-based TDC architecture. Unlike a buffer delay line, the state of the inverter delay line does not directly constitute a thermometer code because an inverter not only introduces delay but also inverts the signal value. This requires a slightly different binary encoding logic used for the TDC output. Basically, if inversion is applied after sampling for every other tap, the output becomes

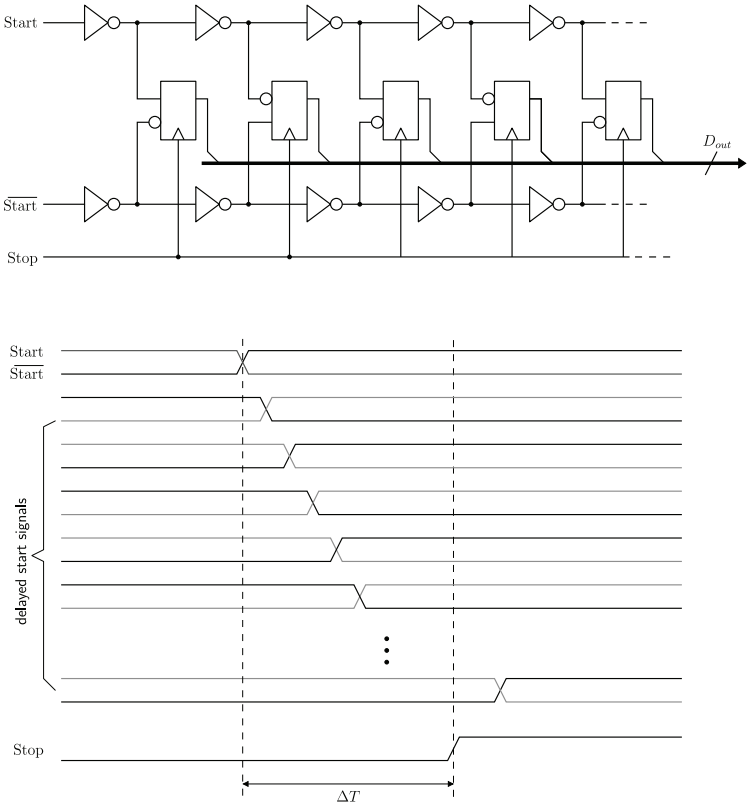


Figure 6.8. An inverter delay-line-based TDC architecture

a thermometer code just as in the case of the buffer delay architecture.

The above delay-line TDC architectures are often collectively categorized as flash TDCs, since the sampling time is performed in parallel at the same time. Obviously, the sampling time with different flip-flops cannot be perfectly aligned in a real implementation due to component mismatches and other issues. Particularly, the high load for the stop signal when clocking all of the sampling flip-flops usually entails the adoption of a buffer tree in its distribution; any unbalance in the buffer tree would introduce skew for the sampling time for the different taps. The mis-alignment of the sampling times to different taps and the delay mismatches among the unit delay cells are major sources of nonlinearity for TDCs of this type.

6.2.3 High-resolution TDC techniques

The time resolution of a flash TDC is limited by the propagation delay of the gate used to realize the unit cells, and in general it cannot go beyond

the delay of one inverter. Though the inverter delay can usually be varied with parametrization, it cannot be arbitrarily reduced due to the self-loading effect when it is sized up.

6.2.3.0.3 Vernier delay-line TDCs To achieve a better time resolution beyond a gate delay, the stop signal can also be passed to a delay chain to create incrementally delayed sampling clock times. The result is a well-known Vernier-line TDC architecture, as shown in Fig. 6.9. The start signal that arrives first is passed to a slower delay line with a larger unit delay. When the stop signal arrives after the start signal, it is passed to the faster delay line with a smaller unit delay. As the stop signal travels down the delay line, it taps one by one the output of each unit cell of the start signal delay line. As long as the delay lines are sufficiently long, the stop signal will eventually catch up with and overtake the start signal when traveling down the respective delay lines. The output of those taps before the catch-up point will yield a high value, while those after the catch-up point will yield a low value. Like the flash TDCs, the overall output is a thermometer-code representation of the input time interval. Its value corresponds to the number of unit cells that have been traversed by the start signal up to the time of the catch-up point instead of the arrival of the stop signal at the TDC.

Suppose τ_1 is the unit delay of the delay line for the start signal and τ_2 is the unit delay of the stop signal delay line, with $\tau_1 > \tau_2$. The time difference for both signals to travel n unit cells along the corresponding delay lines is $\Delta t = n\,(\tau_1 - \tau_2)$. The catch-up point occurs when the time difference becomes large enough to compensate for the input time interval, i.e. $\Delta t \geq \Delta T$. By denoting this output value again as M, we obtain $M = \lfloor \Delta T / (\tau_1 - \tau_2) \rfloor$. The use of two delay lines allows the TDC time resolution to be defined by a difference of unit delays in the two lines, i.e. $T_{\mathrm{lsb}} = \tau_1 - \tau_2$. It thus overcomes the resolution limitation associated with the minimum gate propagation delay. It should be noted that the start and stop signals can be exchanged if we let $\tau_2 > \tau_1$.

It should also be noted that the stop signal in a Vernier TDC does not have to drive a high capacitive load because each flip-flop is clocked by different delayed versions of the signal and thus no additional clock tree is needed to distribute the clock signal. However, like a flash TDC, the hardware complexity and current consumption must increase linearly together with the dynamic range. Compared with the flash architecture, the

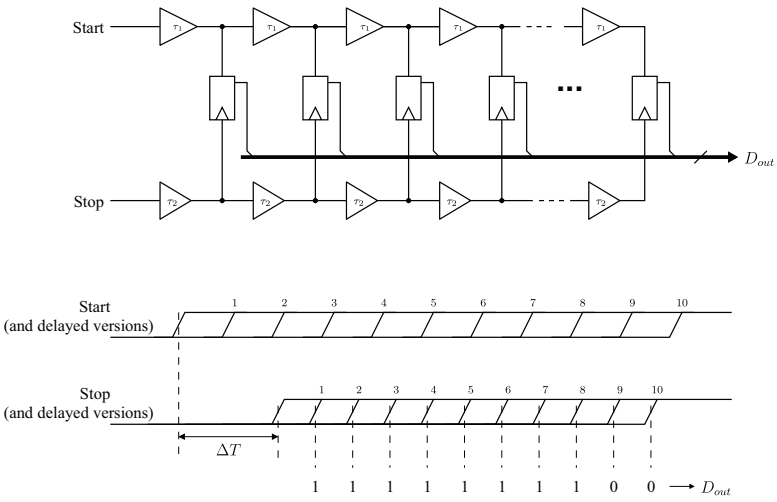


Figure 6.9. A Vernier-line-based TDC.

additional delay line in the Vernier-line TDC introduces additional noise and mismatches. With the sampling times skewed along with the delays, the TDC is also more susceptible to interferences and supply fluctuations. Particularly, the TDC current consumption is usually highly dynamic in time. It tends to cause variations in the supply voltage and thus the unit delay during the time when the sampling is taking place, which would degrade the linearity.

As can be seen, the entire circuit of the delay-line-based TDC can be built using digital cells. such as inverters, buffers and flip-flops, which do not rely on analog voltage for operation. As a result, it has excellent scalability with the CMOS technology. Moreover, it can also readily meet the special requirements of an ADPLL.

6.2.3.0.4 Pulse-shrinking TDCs The principle of a pulse-shrinking TDC is, in a way, similar to that of a Vernier-line architecture. As shown in Fig. 6.10, the main difference is that it has one input signal line that carries both the start and stop timing events, respectively, as the rising and falling transition edges of a pulse. The unit cells in the delay line can be buffers with intentionally uneven delays for the rising and falling edges, with the delay being larger for a rising edge than for a falling edge. Before the arrival of the pulse, the sampling flip-flops in the TDC are all reset to output a low value. As the pulse passes each cell in the delay line, a flip-flop is triggered to sample a high value for its output, while its width shrinks due to the uneven buffer delays. With a sufficiently long delay line, the pulse ceases to propagate further once the falling edge

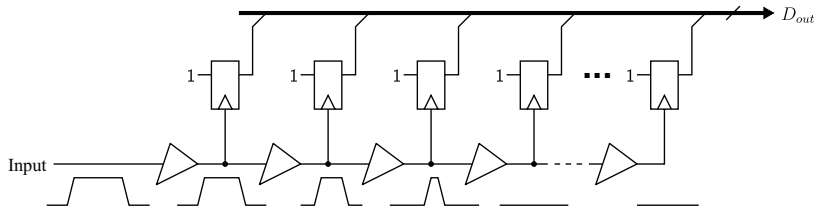


Figure 6.10. Pulse-shrinking delay line TDC.

catches up with the rising edge and reduces the pulse width to zero. As a result, the output of those taps before the vanishing point of the pulse will yield a high value, while the rest would give a low value. The output is a thermometer-code representation of the input time interval, with its value equal to the number of unit cells that the pulse can pass before it flattens out.

Denote the rising edge delay as τ_r and the falling edge delay as τ_f, with $\tau_r > \tau_f$. The pulse width will be reduced by the amount of $T_{\text{lsb}} = \tau_r - \tau_f$ as it traverse each unit cell. Just like a Vernier-line TDC, the pulse-shrinking TDC allows the resolution to go beyond the limitation of the minimum gate propagation delay. Moreover, it saves one delay line in comparison to the Vernier-line architecture. However, applying a pulse-shrinking TDC architecture to an ADPLL architecture is not straightforward considering the special requirements discussed previously.

6.2.3.1 Time amplification

The time resolution of a TDC can be improved if the input time interval can be amplified before it is converted into digital form. Though time amplification tends to be more complicated than voltage amplification, it has been shown to be achievable with a limited linear range. The time amplifier (TA), first published by Abas et al [93], is based on the SR-latch in the metastable region. The same type of TA has been employed to realize a TDC [94].

Fig. 6.11 shows a block diagram of such a time amplifier. Since the TA features a very small linear range of a few ten picoseconds, it usually cannot be utilized directly at a TDC input but instead for the finer stage in a two-stage hierarchical TDC architecture. It should be noted that the gain of such a TA is inaccurate and unpredictable due to component mismatches and PVT variations. As a result, it usually requires gain calibration as in [94].

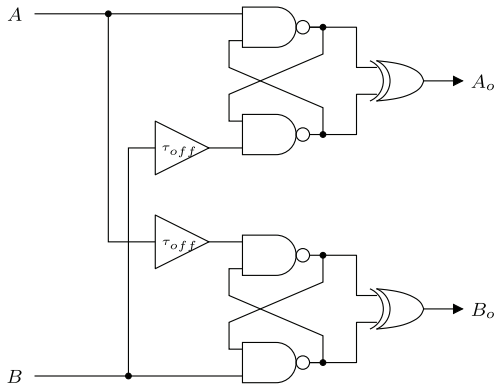


Figure 6.11. A time difference amplifier.

6.2.3.2 Local interpolation

Another technique for achieving high-resolution is local interpolation [95]. The basic idea is illustrated in Fig. 6.12 for a simplest case. The terms V_a and V_b are voltages at the input and output, respectively, of a unit cell in a delay line, while the term V_i is the voltage tapped with a resistive voltage divider. In the given example, we have $V_i = V_a + V_b$. For the delay-line TDC without interpolation, the time delay, τ_d, between V_a and V_b corresponds to one quantization step, which is the TDC time resolution. For every signal transition, it takes a certain amount of time for the voltages to change from rail to rail, and this rising and falling time is typically on the order of a gate delay in a typical delay-line TDC realization with a nanoscale CMOS technology. During the rising or falling edge, the V_a or V_b voltages have a relatively linear relationship with time. Letting $V_a = V_0 + r \cdot t$, it follows that $V_b = V_0 + r \cdot (t - \tau_d)$ during the time when both voltages are in transition. This gives $V_i = V_0 + r \cdot (t - 0.5\tau_d)$, which suggests that the equal voltage interpolation introduces an equal time interpolation. A main concern with passive interpolation is the silicon area, as passive components like resistors are relatively bulky compared with MOS transistors. Meanwhile, the depth of interpolation is limited by the need for an overlapping time for the signal transition edges.

6.2.4 Hardware-efficient TDCs

The delay-line TDCs discussed above comprise one or more feed-forward delay lines with no feedback, and they are often categorized as linear delay-line TDCs. A linear delay-line TDC basically creates timing events that are uniformly spaced in the time domain over the required time in-

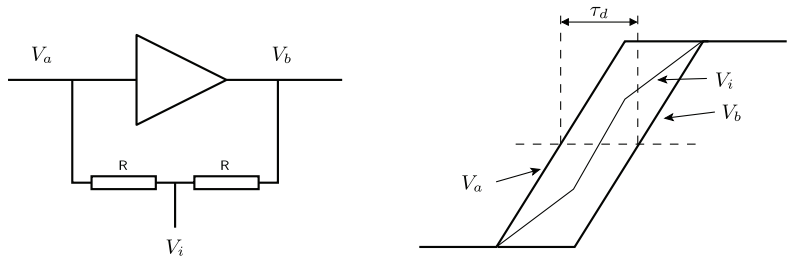


Figure 6.12. Principle of local passive interpolation for a TDC.

terval. Meanwhile, a dedicated unit cell or hardware is needed to create a timing event for every single conversion. The result is that the hardware cost and current consumption grow almost proportionally together with the TDC resolution. It can be prohibitively high for a high-resolution TDC with a large measurement range. In recent years, there have mainly been two different approaches to address the issue and improve TDC hardware or power efficiency. One approach is to connect the delay line as a loop so that each unit cell can be re-used to create multiple timing events spaced in the time domain. Those TDCs employing this approach are called looped delay-line or ring delay-line TDCs. The other approach is to arrange the TDC hierarchically so that it has coarse and fine quantization steps and does not have to measure the entire time interval with full resolution. The resulting timing events do not need to be spaced uniformly over the entire measurement range. The resulting architectures are usually referred to as hierarchical, two-step or coarse-fine TDCs. The two approaches will be examined more closely below.

6.2.4.1 Looped delay-line TDCs

Fig. 6.13 shows a simplified block diagram of a looped delay-line TDC. Compared with a simple delay-line TDC, the delay line in the TDC is connected to form a loop, and there is an additional counter clocked by a signal tapped from the delay line. Before the measurement, the start signal line is at a low level and the loop is in a stable state, with the other input to NAND in a high state. When the start event arrives as a signal rising edge, it passes to the delay line through the NAND gate. Meanwhile, with the start signal line staying high, the loop connection of the delay line forms a ring oscillator, which allows the timing event to circulate until the measurement is done. The counter keeps track of the number of whole cycles the event has circulated. Upon arrival of the stop event, the delay line and the counter are sampled before the TDC is

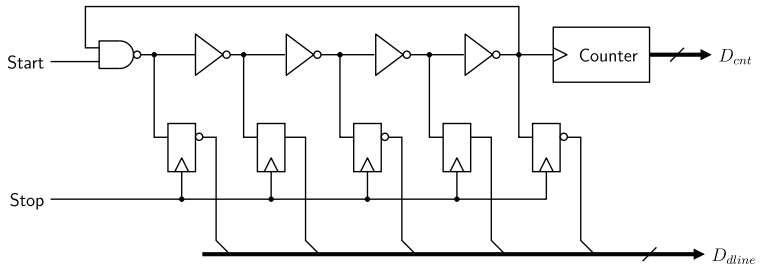


Figure 6.13. A looped delay-line TDC.

prepared for the next measurement. The overall output is a combination of the counter output and the sampled delay-line output as

$$D_{\text{out}} = 2L \cdot D_{\text{cnt}} + D_{\text{dline}}, \tag{6.13}$$

with L the number of delay cells in the delay line.

In the looped TDC, the measurement range is limited only by the counter word length, which can readily be extended without incurring significant hardware cost. The length of the delay line only needs to meet the requirement for minimum clock pulse width set by the counter, regardless of the required measurement range. As a result, the delay line can be very short with a high-speed counter, and the resulting TDC design can be made very compact. Another advantage of the looped TDC is that the delay mismatches among the delay cells make a cyclic contribution to the TDC nonlinearity and do not accumulate over cycles. The main design challenge with a looped delay-line TDC has to do with aligning the counter output with that of the delay line output, which usually requires dedicated control circuitry to ensure the correct alignment in the final result [92]. Another design issue is that the input NAND gate (or MUX in some cases) is always structurally different than rest cells in the delay line and thus contributes to nonlinearity unless re-calibrated in post-processing phase. To assist with calibrating the structural nonlinearity, the looped TDC could be linearly extended [96]. It should be noted that the looped TDC saves on hardware costs when repeatedly using each delay cell to create delayed events, but it does not improve power efficiency because it still measures the entire time interval at full resolution and the resulting number of delayed events is not reduced. The looped approach can in principle also be applied to a Vernier TDC, but usually a relatively high level of complexity is involved [97].

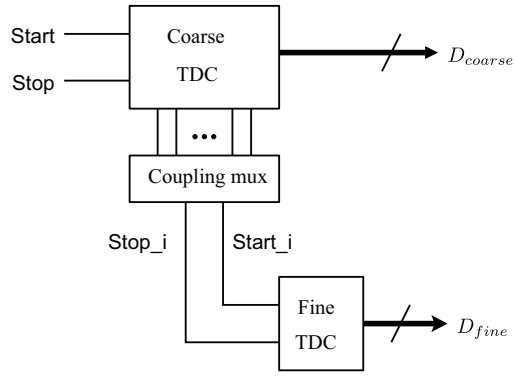


Figure 6.14. A hierarchical TDC approach.

The GRO TDC, as proposed by Helal et al [98], is a special variant of looped delay-line TDC. It feeds the start and stop events not through an NAND or MUX gate, but instead by activating and freezing each inverter stage in the ring oscillator. In principle, such arrangement does not only remove the structural discontinuity associated with the NAND or MUX gate, but also achieves scrambling and noise-shaping effect. However, it is challenging in design to timely and effectively freeze the inverter stages and the ring oscillator internal state till the next start event.

6.2.4.2 Hierarchical TDCs

As shown in Fig. 6.14, a hierarchical TDC usually comprises a coarse TDC and a fine TDC. The coarse TDC covers the entire measurement range, with a relatively coarse quantization step employed to minimize the number of cells needed. The residual time interval, which corresponds to the quantization error from the coarse TDC, is then fed to the fine TDC for fine conversion. The fine TDC features a much finer time resolution with a small measurement range that is just adequate to cover one quantization time step of the coarse TDC. As a result, both the coarse TDC and that fine TDC can be constructed with a greatly reduced number of cells and have a reduced area and power consumption.

The coarse TDC and fine TDC are coupled with a mux with due control logic. The mux and its control logic must be fast enough so that they do not miss the relevant versions of the timing events and pass them to the fine TDC. Typically, additional delay needs to be introduced to the timing events before feeding them to the mux to allow more time for the control logic to be resolved. Meanwhile, the delay introduced by multiplexing needs to be balanced for all the branches, which is another critical design

challenge.

6.2.5 TDC in the ADPLL application

Specific design considerations for a TDC are usually dependent on the target application. As discussed above in subsection 6.1.2, the TDC is the key block for a fine-resolution PDC. Its role is to measure the relevant time intervals to allow for subsequent evaluation of the phase information in the digital domain. Based on the TDC's role in an ADPLL, the following observations can be made regarding special design requirements in an ADPLL for the TDC.

Multi-event capacity: as can be seen, the TDC in the ADPLL is required to have the capacity to simultaneously convert more than one time interval, or so-called multi-event capacity. Since not all TDC architectures have a multi-event capacity, this special requirement is an important consideration when selecting a suitable TDC architecture and its implementation for an ADPLL.

Presence of non-event transition edges: the two signal lines to the TDC are, respectively, the frequency reference signal at a relatively low frequency and an RF signal from the DCO at a much higher frequency. There are typically a number of signal transition edges in the RF signal line during one cycle of the reference clock in the other signal line to the TDC. The time intervals that need to be converted are defined only by the transition edges right next to the reference signal sampling edge, while the other transition edges in the RF line are not relevant or they are non-event transitions. The operation of the TDC must be such that its performance is not compromised by the presence of non-event signal transitions.

Irrelevance of static offset: differentiating the phase signal, ε, to generate the digital frequency signal cancels out the DC component in the phase signal. As a result, common static offset in the TDC transfer function is of no significance.

Relatively small measurement range: in the ADPLL, the maximum time interval to the TDC is limited to one RF cycle. As as result, the required TDC measurement range is relatively small, typically a few hundred picoseconds.

6.3 Summary and discussion

This chapter presented an overview of the PDC in an ADPLL. Common approaches for realizing a PDC were analyzed and compared. The TDC, a key building block for the PDC, has been reviewed in detail. Different TDC architectures and techniques were studied and their advantages and disadvantages analyzed. Particular design considerations for a TDC in an ADPLL were examined.

7. Prototypes and experimental results

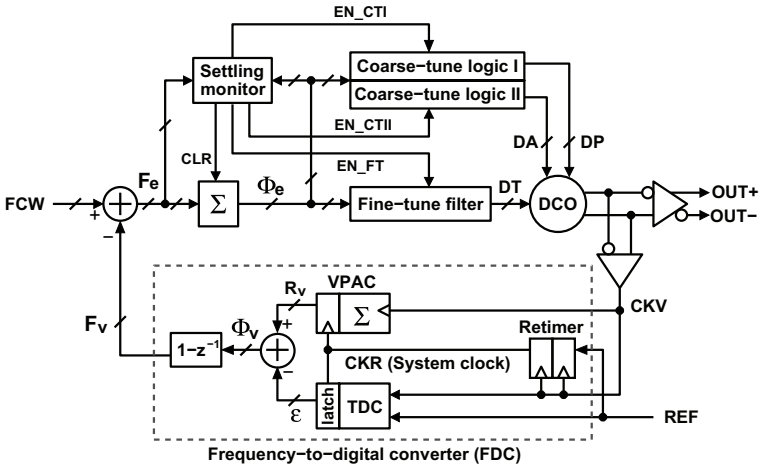
In this chapter, three ADPLL-based frequency synthesizer prototypes are presented. The first prototype is designed for the 2.4-GHz ISM band, with a focus on low power consumption [12]. The second is a wideband frequency synthesizer targeted for spectrum sensing in cognitive radios [8]. The third is another wideband frequency synthesizer designed for demonstrating architectural improvement and a fast frequeny calibration technique [13]. The designs are part of original research work done for this thesis, previously published in several articles [8, 12, 13].

7.1 A 2.4-GHz ISM-band ADPLL frequency synthesizer

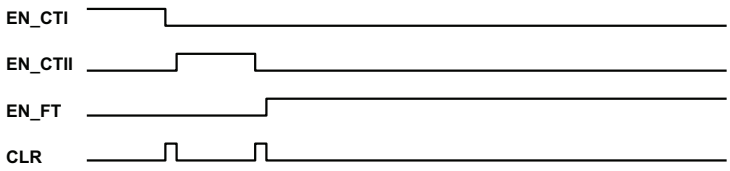
This section presents a 2.4-GHz ADPLL design. It is based on a similar architecture proposed in Staszewski's original work [4]. Circuit techniques are introduced to lower power consumption, simplify circuit implementation and improve the performance and reliability of the ADPLL. In particular, a high-speed topology is employed for the variable phase accumulator (VPA) to allow for reliable handling of the RF signal with its fully digital realization. A delay-based power saving circuitry is employed for the core of the time-to-digital converter (TDC) to operate it at a minimal duty cycle with about 95% power reduction for the circuit block. Subsection 7.1.1 introduces the top-level architecture of the ADPLL, and subsection 7.1.2 describes the circuit implementation details, with the experimental results provided in subsection 7.1.3.

7.1.1 Top-level architecture

Fig. 7.1 (a) shows a top-level schematic of the implemented ADPLL. The digitally controlled oscillator (DCO) is an on-chip LC oscillator, whose out-



(a)



(b)

Figure 7.1. The ADPLL: (a) a top-level schematic and (b) an illustration of the settling control signals.

put frequency can be tuned using three digital control words. *DP* and *DA* are used for coarse tuning, while *DT* is used for fine tuning. The feedback path is a frequency to digital converter (FDC) that detects and digitizes frequency information of the RF signal, *CKV*, generated by the DCO. The frequency digitization provided by the FDC is performed in two steps. First, the phase information of the *CKV* is measured and digitized by counting its cycles, with the result sampled at the frequency of the reference signal, *REF*. The phase counting is jointly done by the VPA with a resolution of one *CKV* cycle and the TDC providing fractional correction to improve the counting resolution. The frequency information is then computed by taking the derivative of the measured phase in the digital domain. The reference signal retiming, which is done by using two flip-flops shown as part of the FDC block, synchronizes the reference clock with the RF signal. This allows for synchronous digital realization with automated design flow for digital circuits (mainly the VPA) that cross the frequency domains of the RF signal and reference signal.

The rest of the loop circuitry is composed of purely digital logic blocks

operating in the clock domain of the retimed frequency reference, *CKR*. Following the phase accumulator, the forward path diverges into three branches for coarse and fine frequency tuning of the DCO. Based on the phase and frequency errors, the settling process monitor controls the activity transitions from one branch to the other and clear the phase accumulator content during each transition. This is illustrated with a timing diagram of the relevant control signals in Fig. 7.1 (b). During the initial phase of the settling process, the two coarse-tune logic branches operate successively to calibrate the DCO output frequency. The fine-tune branch becomes active after the frequency calibration to suppress the FDC quantization noise in the phase error and tunes the DCO for final frequency settling and frequency tracking.

The *FCW* defines the target frequency in units of the reference frequency. The length of its fractional part determines the frequency resolution of the ADPLL. With a 24-bit *FCW* fractional part and a selected reference frequency of about 12 MHz, the implemented ADPLL has a fine frequency resolution of about 1 Hz. As an average concept, this frequency resolution is independent of the DCO frequency resolution. The latter is a measure of the DCO frequency quantization that has undesirable effects of generating additional phase noise at the ADPLL output [55] and increasing nonlinearity in the loop dynamics. During the fine-tune phase, the DCO with a digital $\Sigma\Delta$-modulator at its interface for capacitance dithering provides an effective frequency resolution as small as 1 kHz to minimize the effects of DCO frequency quantization.

For a low-power implementation, the focus of the circuit techniques is on circuit blocks that deal with the RF signal, which include the DCO, TDC and VPA circuit blocks. These circuit blocks present key design challenges and, when combined, account for predominant power consumption of the ADPLL. With the reference frequency lower than the RF output frequency by roughly two orders of magnitude, the power consumption of the digital circuitry working in the reference frequency domain is relatively small for a typical implementation. As an example, all of the digital circuitry in the reference frequency domain only accounts for about 20% of the total power consumption of the implemented ADPLL in this work.

7.1.2 Circuit implementation

This subsection provides implementation details about the ADPLL, with an emphasis on the new circuit techniques and design considerations for low power consumption.

7.1.2.1 Digitally controlled oscillator

As an RF signal generator, the DCO is one of the most power-hungry circuits in the ADPLL. For a power-efficient implementation, the primary focus has been on optimizing the non-power factors for phase noise performance so that a good phase noise performance can be achieved with a relatively low power consumption. This involves minimizing passive component loss, reducing active device noises and improving its noise immunity to control and supply voltage lines.

As shown in Fig. 7.2, the DCO implementation is based on a differential LC oscillator. The LC tank comprises a center-tapped inductor and three capacitor banks. Digital frequency control is performed through the capacitor banks, each of which consists of a number of unit cells that can be individually switched between low and high capacitance states. An array of tail transistors is used for digital current control to allow for a trade-off between power consumption and phase noise performance. The interface logic implements circuit techniques to boost the DCO performance during the fine-tune stage of the ADPLL.

The 2-nH inductor chosen from the technology library has a high quality factor of about 17 at 2.4 GHz. The three varactor banks are parameterized according to Table 7.1 for digital frequency control. The Cbank_P is the largest capacitor bank intended for coarse frequency tuning. Unit cells in this bank utilize a differential structure of two metal-oxide-metal (MOM) capacitors, as shown in Fig. 7.3 (a). This switched capacitor structure provides good frequency stability and noise immunity with respect to digital control line voltages, which is essential when the use of on-chip voltage regulators is avoided out of consideration for power consumption. For low tank loss, the transistor switches, particularly M0, are sized to minimize the on-resistance. Moreover, this large capacitor bank is placed close to the inductor coil in the physical layout and connected to the latter using wide multi-layer metal paths to ensure minimal loss introduced by wiring. The CBank_A and CBank_T utilize NMOS varactors, as shown

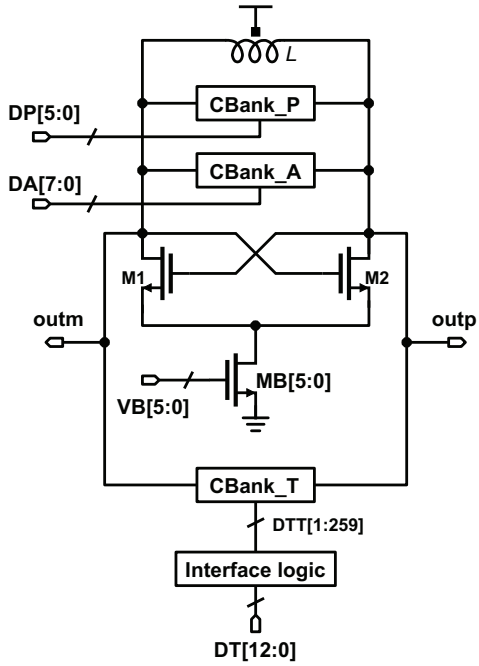


Figure 7.2. Digitally controlled oscillator.

in Fig. 7.3 (b), for smaller frequency steps that cannot be achieved using MOM capacitors. For phase noise consideration, the channel lengths for the cross-coupled gain stage transistors and the tail transistors are larger than the minimum value to avoid the excessive flicker noise that is associated with short-channel effects. For the cross-coupled gain transistor pair, the channel length was selected to be 0.12 μm, slightly larger than the minimum. It was observed that the reduction of flicker noise with slightly larger channel length outweighs the negative impact of increased parasitic capacitance. It was also observed that the phase noise performance eventually degrades with further increase of the channel length. In other words, the selected channel length was a good compromise between the transistor flicker noise and the parasitic capacitance. The tail transistors are sized to have a long channel of 1 μm. In simulations, clear increase in the flicker noise was observed with a smaller channel length, while little improvement could be observed with further increase of the channel length.

The binary control word, *DT*, comprises an 8-bit integer part and a 5-bit fractional part. In the interface logic, the integer part is converted to a thermometer code with pseudo-random scrambling to ensure monotonicity and good linearity of frequency tuning. The fractional part is fed to

Table 7.1. Key design parameters of a DCO capacitor/varactor bank

Capacitor bank	Weighting	Frequency step	Frequency range
Cbank_P	6-bit binary	~8 MHz	> 500 MHz
Cbank_A	8-bit binary	~450 kHz	> 100 MHz
Cbank_T	259-bit unitary	~20 kHz	> 5 MHz

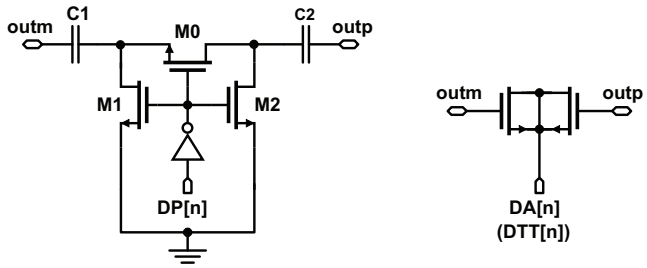


Figure 7.3. Unit cell structure in (a) CBank_P, (b) CBank_A and CBank_T.

a 2nd-order digital sigma-delta modulator (SDM) clocked with a divide-by-4 high-frequency signal derived from the DCO output. Based on simulations, the high clock frequency minimizes the SDM noise effect while still allowing the SDM to only consume about 0.15 mW. The SDM output controls three unit varactors in the CBank_T and performs high-speed capacitance dithering to achieve a corresponding fractional resolution of 625 Hz from the unit frequency step of 20 kHz. The fine frequency resolution of the DCO is essential to prevent the DCO frequency quantization from degrading the overall phase noise performance of the ADPLL.

7.1.2.2 Variable phase accumulator

The VPA is one of few circuit blocks located at the boundary between the frequency domains of the RF signal and reference signal. It is highly desirable for this block to be realized as a synchronous digital circuit instead of an asynchronous circuit so that it can fit with the rest of the digital circuitry for an automated digital design flow without complicating the overall implementation of the ADPLL. This adds to the challenge that the VPA needs to perform high-speed cycle counting of the RF signal, *CKV*. The high frequency of the *CKV* essentially rules out the choice of a straightforward VPA realization using a binary counter followed by a sampling register, as shown in Fig. 7.4 (a). With combinational logic introducing additional delay to the timing-critical paths, a binary counter has a rela-

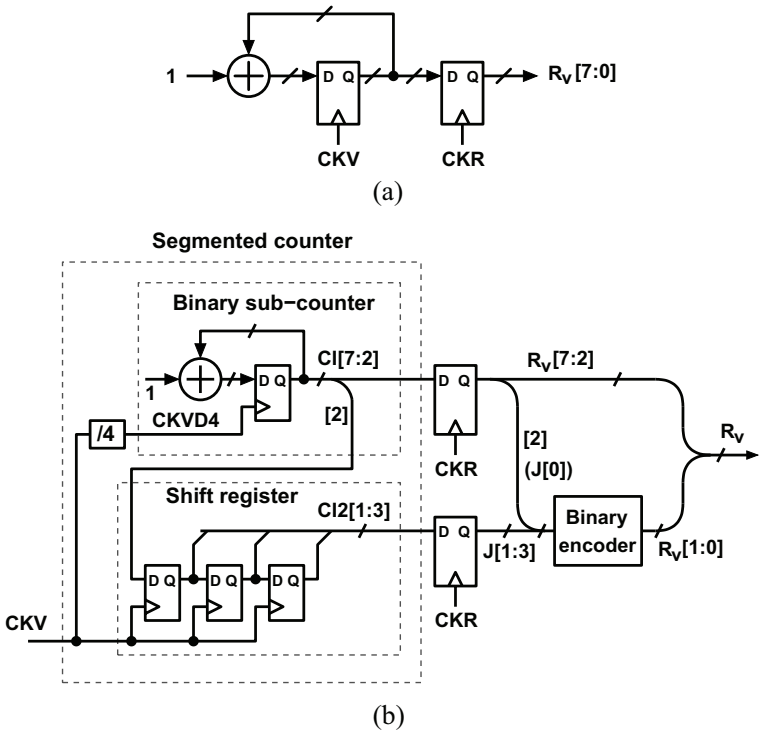


Figure 7.4. VPA: (a) a basic realization, and (b) implemented high-speed topology.

tively low counting speed. Placing it directly in the frequency domain of the *CKV* would overstrain its speed capacity. Existing circuit techniques can be used to improve the counting speed of a binary counter [99–101]. However, these techniques cannot entirely eliminate the combinational logic in a synchronous binary counter, which limits the counting speed and leads to unreliable operation at high *CKV* frequencies.

To overcome the above issue, a high-speed structure, as shown in Fig. 7.4 (b), is proposed for the VPA realization. It utilizes a segmented counter, which comprises a divide-by-4 block, a 6-bit binary counter and a shift register, to count the cycles of the RF signal, *CKV*. After sampling by the retimed reference signal, *CKR*, the counting result is converted into a binary form by encoding its non-binary part in the low-frequency reference clock domain. The encoding logic, as given in Table 7.2, can be derived by simply comparing the segmented counter with an equivalent binary counter [5]. The 8-bit VPA word length is chosen based on the requirement that the counter modulus needs to be larger than the maximum ratio of the RF output frequency to the reference frequency to ensure correct functioning of the FDC.

Table 7.2. Binary encoding logic in VPA

J[3:0]	R_v[1:0]
0000	11
0001	00
0011	01
0111	10
1111	11
1110	00
1100	01
1000	10
others	don't care

Using a segmented counter instead of a single binary counter, the VPA effectively keeps the combinational logic associated with a binary counter away from the high-frequency domain of the *CKV*. The binary sub-counter is now placed in a lower frequency domain where its operation speed is not critical. The circuitry in the *CKV* domain is reduced to purely sequential components without delay introduced by combinational logic, and it is thus capable of the maximum-speed operation achievable with a synchronous digital circuit. This high-speed capacity makes it possible to reliably count the RF signal up to a frequency of 3 GHz without involving any custom design, which could not be achieved if a straightforward realization were to be utilized. Furthermore, it consumes less power due to reduced circuitry in the high-frequency domain. It should be noted that the frequency divider is also needed for clocking the digital SDM in the DCO interface logic and it is shared here without using an additional divider.

7.1.2.3 Time-to-digital converter

The phase digitization of the RF signal performed by the VPA features a resolution of one full cycle, which translates into large quantization noise in the detected frequency and phase errors in the forward path and limits the ADPLL performance. The function of the TDC is to improve the resolution of the phase digitization with a sub-cycle time resolution and reduce the quantization noise. As one of few circuit blocks that directly work with an RF signal and still require semi-analog design, the TDC can consume a large amount of power especially when designed for a high resolution [91].

Fig. 7.5 (a) shows the TDC core implementation, which utilizes a pseudo-

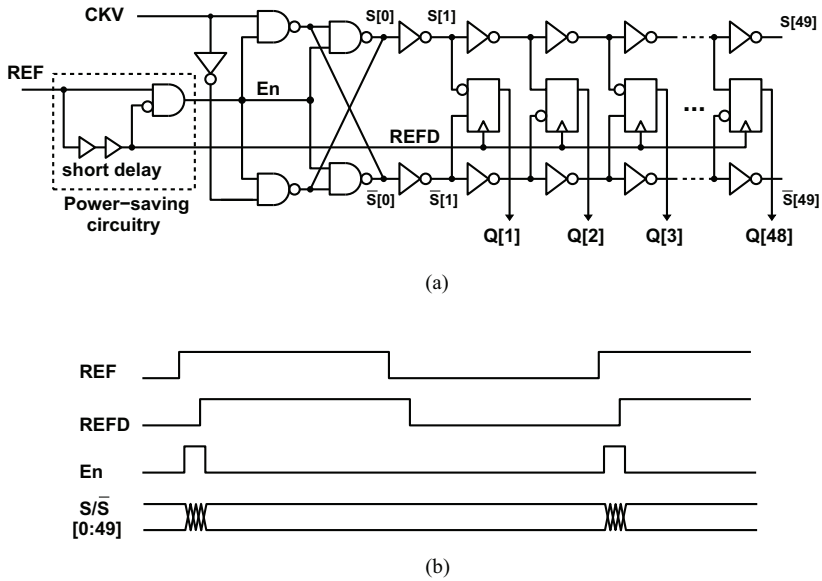


Figure 7.5. TDC core: (a) a schematic and (b) signal illustration of a power saving technique.

differential inverter-chain-based architecture [102], with an effective power-saving technique. The adopted architecture is relatively simple to implement compared with a variety of other TDC architectures reported in the existing literature for ADPLL application [89, 94, 95]. It provides a time resolution of one inverter delay, which is around 20 ps for a typical implementation in 65-nm CMOS. The TDC resolution determines the minimum in-band phase noise achievable at the ADPLL output. This in-band phase noise level can be estimated as [53]

$$\mathcal{L}_{QTDC} = \frac{\pi^2}{3} \cdot \left(\frac{\Delta t_{\mathrm{tdc}}}{T_{\mathrm{ckv}}} \right)^2 \cdot \frac{1}{f_{\mathrm{ref}}}, \tag{7.1}$$

where Δt_{tdc} is the TDC time resolution, T_{ckv} the *CKV* period, and f_{ref} the reference frequency. With the selected reference frequency, the 20 ps TDC time resolution corresponds to an in-band phase noise contribution of about -92 dBc/Hz, a value sufficiently low for target applications in the ISM band. The number of taps selected was 48, which is the number needed to cover a full *CKV* cycle with PVT variations taken into account.

To save on power consumption, the reference, *REF*, passes through a short delay line before clocking the sampling flip-flops. The inverter chains are activated by every rising edge of the *REF* and deactivated by that of its delayed replica, *REFD*. The delay line was implemented to give a slightly longer delay than that of the *CKV* inverter chains. As illustrated with the

timing diagram in Fig. 7.5 (b), this allows the inverter chains to be active only for a small time window at every sampling moment, leading to a substantial reduction in the average power dissipation. Since the *REF* is at a much lower frequency than the *CKV*, power overhead of the delay line is negligible. The technique reduces the power consumption of the TDC core roughly from 4 mW to as small as 0.2 mW. The constant delay introduced by the short delay line amounts to a fixed offset in the detected *CKV* phase information, which is of no significance for the ADPLL. The delay line also introduces a timing jitter to the reference signal, which on the one hand contributes to the in-band phase noise of the ADPLL output, while on the other hand it provides the dithering effect that suppresses in-band spurs arising from TDC quantization. In most cases, the delay-line jitter tends to be much smaller than the 20-ps TDC quantization step, and thus its effect is insignificant. Like the TDC quantization noise [53], the delay line jitter contribution to the ADPLL in-band phase noise can be quantified in a similar way as

$$\mathcal{L}_{DJ} = \left(\frac{2\pi\sigma_j}{T_{\text{ckv}}}\right)^2 \cdot \frac{1}{f_{\text{ref}}}, \tag{7.2}$$

where σ_j represents the RMS jitter value. Based on the simulations, the implemented delay line has a RMS jitter value of about 2 ps. Considering a *CKV* frequency of 2.4 GHz and a reference frequency of 12.3 MHz, the corresponding $\mathcal{L}_{DJ}$ should be approximately -101.3 dBc/Hz. According to the simulations, a 2-ps jitter appears to lower the worst-case quantization spurs by approximately 2 to 3 dB.

The TDC core output is decoded to give binary representations of the *CKV*'s rising and falling edge delays relative to the sampling reference edge. Denoted as $\Delta t_r/\tau_{\text{inv}}$ and $\Delta t_f/\tau_{\text{inv}}$, respectively, in Fig. 7.6, the two values are both in units of one inverter delay. The normalization block is used to derive the corresponding value of the rising edge delay in units of one *CKV* period, T_{ckv}. It involves an arithmetic division carried out as a combination of inversion and multiplication. The instantaneous *CKV* period, $\tilde{T}_{\text{ckv}}/\tau_{\text{inv}}$, calculated as twice the difference between the rising and falling edge delays, is inaccurate and fluctuates over time due to the time quantization and particularly the inevitable deviation of the *CKV* from an ideal duty cycle of 50%. In this implementation, accuracy is improved by using a single-pole infinite-impulse-response (IIR) filter for its superior efficiency over a finite-impulse-response (FIR) filter in terms of the amount of circuitry and power dissipation. By placing the filter after the inversion,

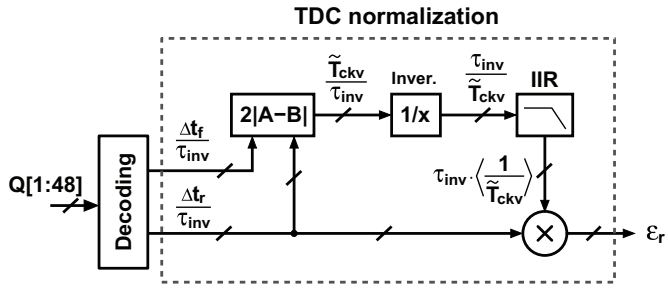


Figure 7.6. TDC normalization with a simplified implementation.

instead of between the period calculation and inversion [102], the inversion has a shortened word length and consequently be implemented using a small lookup table with significantly reduced complexity. The output of the normalization block can be expressed as

$$\varepsilon_r = \Delta t_r \cdot \left\langle \frac{1}{\tilde{T}_{\mathrm{ckv}}} \right\rangle \approx \frac{\Delta t_r}{T_{\mathrm{ckv}}}, \tag{7.3}$$

where the approximation is partially introduced by exchanging the filtering and inversion operations. It can be shown that this approximation is small and has an insignificant effect that only slightly increases the level of TDC quantization noise.

As previously discussed in Chapter 6, operation of the TDC along with a VPA suffers from the glitch or ambiguity issue [12]. In this design, a correction circuit, as shown in Fig. 7.7, is placed in the forward path to correct the detected frequency error before it is used by a subsequent circuitry. Its operation is activated after the initial frequency calibration with the coarse-tune logic. During the initial frequency calibration, where the deviation of the ADPLL output frequency from the target frequency is large, the effect of TDC ambiguity errors is insignificant. After the frequency deviation is reduced by the frequency calibration, the correct value of F_e should be small, and it should be much smaller than 1 during the fine-tune phase. The TDC ambiguity errors would thus manifest themselves as a train of impulses of a large magnitude. By comparing F_e with the predefined threshold values, it detects both the occurrence and signs of the ambiguity errors. Accordingly, the frequency error, F_e, is incremented or decremented by one to cancel out the TDC ambiguity error.

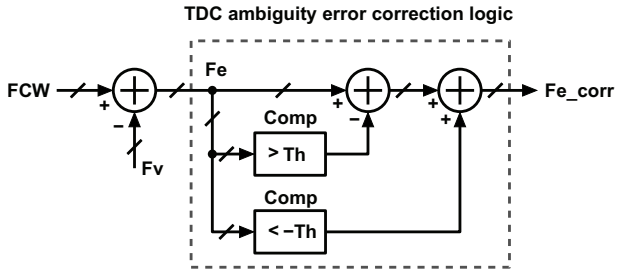


Figure 7.7. TDC ambiguity correction circuit ($T_h = 0.5$ after frequency calibration).

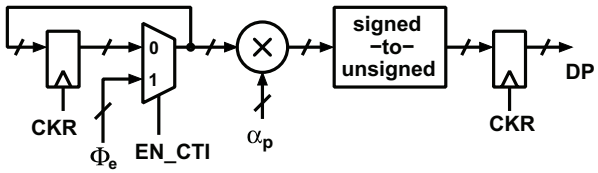


Figure 7.8. Coarse-tune logic branch for the DCO control word DP.

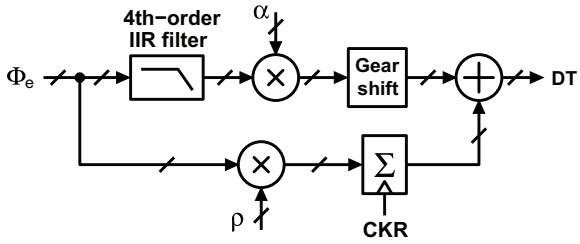


Figure 7.9. Fine-tune digital loop filter.

7.1.2.4 Other circuit blocks

Fig. 7.8 shows the coarse-tune logic for the control word *DP*, where an attenuation factor determines the loop gain and was selected for fast settling. Signed-to-unsigned conversion is implemented by a simple MSB inversion. The coarse-tune logic for the control word *DA* is the same except for different parameters.

As shown in Fig. 7.9, the fine-tune digital loop filter is a fifth-order structure [4], with adjustable filter coefficients. When the frequency approaches the target frequency during the final settling process, the gear shift block seamlessly scales down the loop bandwidth for better suppression of the FDC quantization noise. With default loop filter parameters, the final ADPLL bandwidth is about 40 kHz. The high-order filter with gear shifting allows for sufficient attenuation of the FDC quantization noise without excessively slowing down the frequency settling.

In addition, a two-point frequency modulation scheme, like in Staszewski's work [4], has been implemented with the ADPLL as shown in Fig. 7.10.

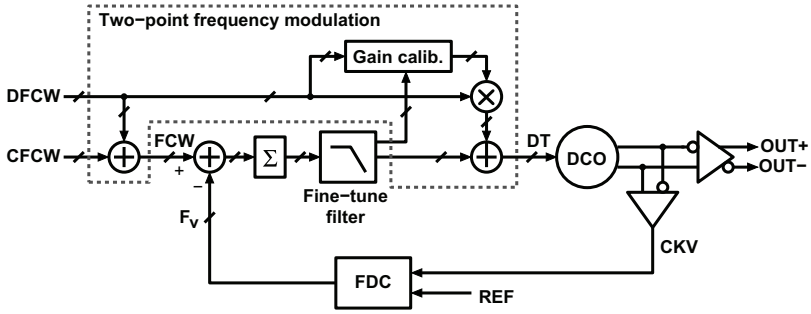


Figure 7.10. Two-point modulation implemented with the ADPLL.

Assisted by the least-mean-square (LMS)-based gain calibration [103] to compensate for PVT variation in the DCO frequency tuning gain, the ADPLL is effectively an all-pass filter for the data frequency control word *DFCW*, allowing for a truly wideband direct frequency modulation with essentially no constraint from the loop bandwidth.

7.1.3 Implementation results

Fig. 7.11 shows a die micrograph of the ADPLL fabricated in a 65-nm CMOS. It has an active area of 0.24 mm^2. With a 1.2-V supply, the total power consumption of the ADPLL can be set between 8 and 12 mW; the DCO power consumption can be tuned over the range of 5 to 9 mW maintaining sufficient oscillation amplitude for the proper operation of the ADPLL. The TDC core dissipates about 0.2 mW and all of the digital logic has a combined power consumption of about 2.6 mW. The measured frequency locking range was 2.29–2.92 GHz. The circuit was tested over the frequency range, with the power consumption respectively set to about 8 mW and 12 mW. For the two different levels of power consumption, the worst-case phase noise at the 1-MHz frequency offset was -112 dBc/Hz and -120 dBc/Hz, respectively. The spur levels were not measurably different in the two different cases of power consumption. The worst in-band spur level shown in the phase noise measurements for the target ISM band was -61 dBc measured with a resolution bandwidth is 1 kHz. The in-band phase noise level is about -81 dBc/Hz. It is about 6.6 dB higher than the estimated contribution from the TDC quantization. The difference is due to contributions from other sources including the DCO, the reference clock as well as the delay lines within the TDC. The integrated phase noise was measured to be 1.7° rms over the frequency range from 1 kHz to 10 MHz.

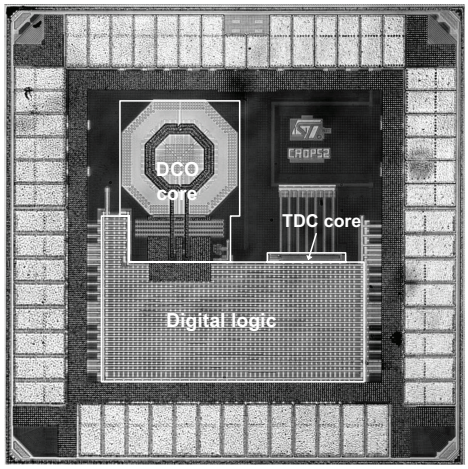


Figure 7.11. Die micrograph.

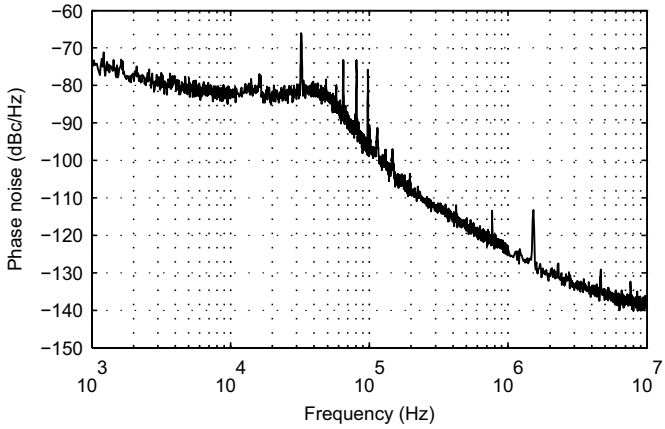


Figure 7.12. Measured phase noise at 2.45 GHz.

Fig. 7.12 shows an example of the measured phase noise spectrum at 2.45-GHz output frequency with 12-mW power dissipation. Fig. 7.13 shows the output spectrum over a 1-GHz span for the same case. The far-off spurs were below −69 dBc. It was found that the leakage of high-order reference harmonics due to imperfections in the physical layout was the dominant source of far-off spurs that appeared in the spectrum. The majority of spurs remained in the spectrum when the digital part of the ADPLL was powered off, leaving the oscillator free running with the reference signal fed to the test chip. The measured ADPLL output frequency settling process of the ADPLL is shown in Fig. 7.14. With limited bandwidth and detection speed, the measurement equipment was not able to faithfully capture the output frequency during the coarse-tune phase due to its large deviation from the target frequency and rapid change over time.

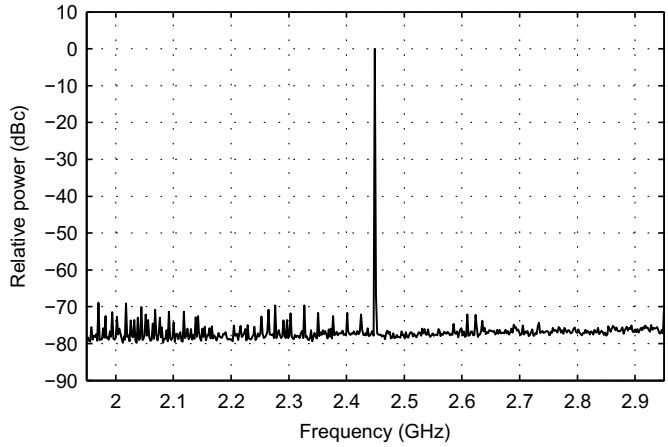


Figure 7.13. Measured output spectrum at 2.45 GHz.

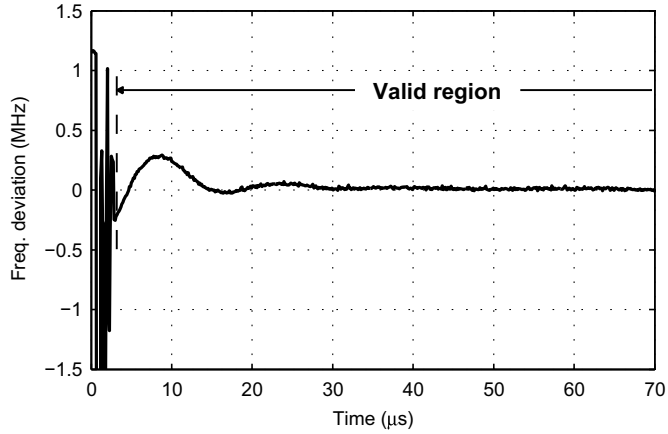


Figure 7.14. Measured frequency settling of the ADPLL output.

However, the final settling was in good agreement with the simulated results, both of which showed a maximum settling time of about 30 μs. Fig. 7.15 shows the measured output frequency deviation from a center frequency of 2.45 GHz in the time domain when a binary frequency shift keying starts with randomly generated symbols. The frequency deviation and symbol rate were arbitrarily selected to be about 1 MHz and 2 Mbps, respectively. As can be seen, the gain calibration settled within a small number of symbols, after which the modulation was little disturbed by the limited loop bandwidth. Fig. 7.16 shows a measured output spectrum when the frequency modulation with randomly generated symbols was on.

The measurement results are summarized in Table 7.3. The 1-Hz frequency resolution corresponds to 24-bit fractional part of FCW for the

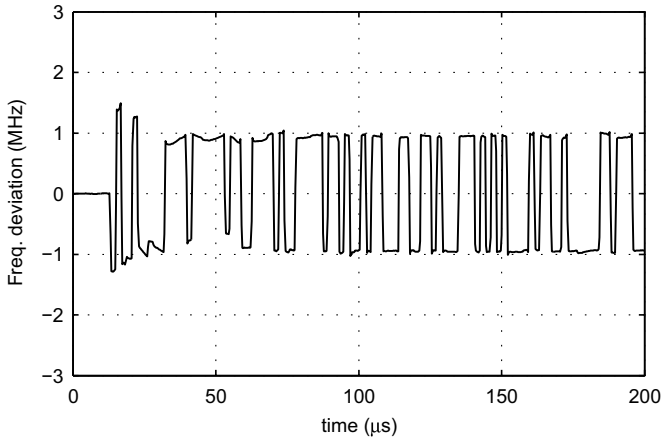


Figure 7.15. Measured ADPLL output frequency when frequency modulation starts.

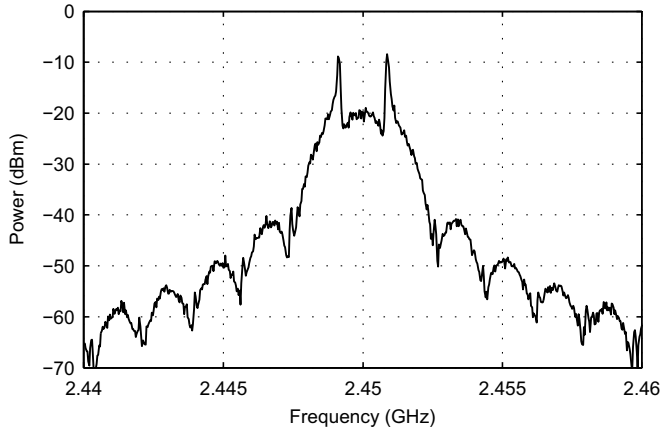


Figure 7.16. Measured output spectrum with frequency shift keying.

given reference frequency. It was confirmed in simulations though not measured directly due to insufficient measurement equipment. Table 7.4 provides a comparison of this work with relevant prior-art PLL implementations targeting the 2.4-GHz ISM band. It can be seen that the ADPLL designed in this work leverages the miniaturized 65-nm CMOS process and achieves low power consumption and a small die area with good performance.

7.2 A 3-6GHz ADPLL frequency synthesizer

Frequency synthesizers for the sensor units in cognitive radios need to cover a wide frequency range that can stretch over a multitude of existing licensed and unlicensed bands. Meanwhile, low power consumption and

Table 7.3. Performance summary

Output frequency range	2.29–2.92 GHz	
Frequency resolution	1 Hz	
Loop bandwidth	40 kHz	
Frequency modulation range	± 2.5 MHz	
Power consumption	8 mW	12 mW
Phase noise @ 1 MHz	−112 dBc/Hz	−120 dBc/Hz
In-band phase noise	−81 dBc/Hz	
In-band spur	−61 dBc *	
Far-off spur	−69 dBc	
Settling time	≤ 30 μs	
Supply voltage	1.2 V	
Reference frequency	12.3 MHz	
Active area	0.24 mm^2	
Technology	65-nm 1P6M CMOS	

* From phase noise measurements with a 1-kHz resolution bandwidth.

fast settling are important considerations. This section presents a wideband frequency synthesizer for such sensor units. It is capable of covering an octave (2:1) band from 3 to 6 GHz. Thereafter, all lower bands can be covered easily by using divide-by-2^N circuits. The frequency synthesizer is based on an ADPLL with adaptive frequency calibration (AFC). For wideband frequency tuning, it employs a ring oscillator with an LC tank introduced to extend the tuning range and decrease power consumption. The top-level architecture of the frequency synthesizer is described in subsection 7.2.1, while subsection 7.2.2 discusses the implementation of the frequency synthesizer. Experimental results are given in subsection 7.2.3.

7.2.1 Top-level architecture

Fig. 7.17 shows the top-level architecture of the ADPLL frequency synthesizer. The *FCW* defines the target frequency. The RF output is generated by a digitally controlled oscillator (DCO). The feedback path is a frequency-to-digital converter (FDC) that converts frequency of the RF signal into digital form. Particularly, the *FCW* and FDC output, f_v, are two fixed-point numbers that respectively represent the target frequency and the detected RF output frequency, both of which are normalized by the reference frequency. The frequency error is computed by subtracting

Table 7.4. Comparison with relevant prior arts

Parameter	ISSCC'04 [104]	JSSC'04 [105]	JSSC'06 [106]	JSSC'07 [107]	ISCAS'08 [108]	JSSC'08 [109]	JSSC'08 [110]	**This work**
Output frequency	N/A	2.4–2.48 GHz	1.6–2.0 GHz	2.4–2.5 GHz	2.2–2.6 GHz	max. 2.24 GHz	2.4–2.5 GHz	**2.29–2.92 GHz**
Loop bandwidth	N/A	400 kHz	400 kHz	730 kHz	400 kHz	142 kHz	975 kHz	**40 kHz**
In-band phase noise	−86 dBc/Hz	−96 dBc/Hz	−98 dBc/Hz	−101 dBc/Hz	−81 dBc/Hz	−79 dBc/Hz †	−103 dBc/Hz	**−81 dBc/Hz**
Far-off phase noise	N/A	−121 dBc/Hz @3MHz	−118 dBc/Hz @1MHz	−124 dBc/Hz @3MHz	N/A	−108 dBc/Hz @1MHz †	−125 dBc/Hz @3MHz	**−120 dBc/Hz @1MHz**
Fractional spur	−62 dBc	−40 dBc	−70 dBc	−47 dBc	N/A	−45 dBc	−64 dBc	**−61 dBc**
Settling time	50 μs	N/A	N/A	35 μs	N/A	N/A	N/A	**30 μs**
Reference frequency	26 MHz	48 MHz	14.3 MHz	12 MHz	20 MHz	185.5 MHz	12 MHz	**12.3 MHz**
Power consumption	37.5 mW §	67 mW	29 mW	38 mW	22 mW	14 mW	27.1 mW	**12 mW**
Core area	N/A	1.6 mm^2	2 mm^2 ‡	4.8 mm^2 ‡	2.2 mm^2 ‡	0.7 mm^2	4.84 mm^2 ‡	**0.24 mm^2**
Technology	0.13 μm	0.18 μm	0.18 μm	0.18 μm	0.18 μm	0.13 μm	0.18 μm	**65 nm**

§ Total transmitter power consumption. † Estimated based on the phase noise plot. ‡ Full chip size including pads.

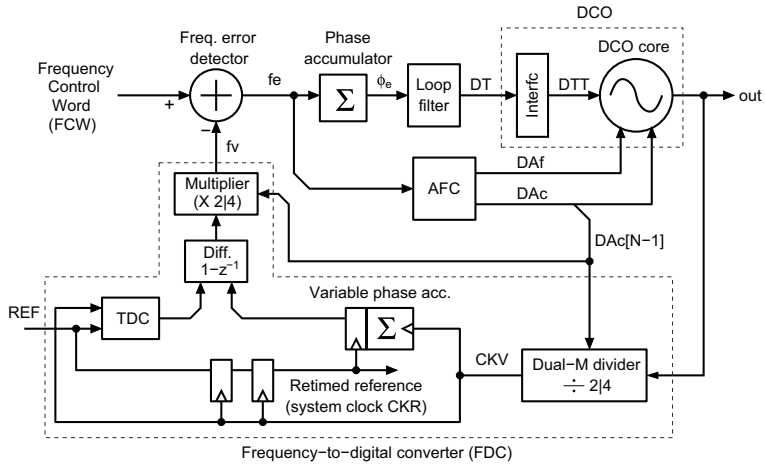


Figure 7.17. Top-level architecture of the ADPLL frequency synthesizer

the two numbers from one another. Following the frequency error detector, a phase accumulator and loop filter close the loop for the phase and frequency lock. To achieve fast frequency settling over a wide frequency range, an adaptive frequency calibration (AFC) is employed to assist in frequency acquisition. Each time a new target frequency is requested, a frequency calibration is first performed, during which time the AFC block is activated to tune the DCO and set its output frequency close to the target frequency. The phase accumulator and loop filter are set to output a constant *DT* value at the middle of its range during frequency calibration, and they are activated afterwards to finalize the frequency settling and perform frequency tracking.

7.2.2 Circuit Description

The difference between this ADPLL design and our first design presented in section 7.1 has to do with the FDC, the AFC and the DCO blocks, where new features were implemented. Other blocks remain essentially the same as in the previous design. This subsection discusses the implementation of major building blocks that have new features.

7.2.2.1 Frequency-to-digital converter

With the FDC, a dual-modulus frequency divider divides the RF signal by a factor of 4 or 2, depending on the signal frequency. This ensures a narrowed frequency range of roughly 1 to 2 GHz at the divider output so that the resulting signal can be reliably handled by the VPA and does not re-

quire an excessively long chain of inverters and flip-flops in the TDC. The different division ratios are compensated for by a dual-modulus multiplier at the FDC output, keeping the FDC gain constant. The TDC is based on an inverter-chain structure [103], while the VPA utilizes a high-speed topology as described in section 7.1.2.2. The FDC frequency resolution is limited by the TDC time resolution and can be expressed as

$$f_{\mathrm{res}} = T_{\mathrm{res}} \cdot f_{\mathrm{ref}} \cdot f_{\mathrm{out}}, \tag{7.4}$$

where T_{res} represents the TDC time resolution, f_{ref} is the reference frequency, and f_{out} denotes the ADPLL output frequency. With the inverter-chain structure, the TDC has a time resolution of about 20 ps in the 65-nm CMOS. For a reference frequency of 12.3 MHz and an RF output of 4 GHz, it gives an FDC frequency resolution of about 1 MHz, in contrast with the ADPLL frequency resolution of roughly 1 Hz, which corresponds to a 24-bit fractional part of the *FCW* adopted for implementation. Quantization noise due to the relatively coarse FDC resolution dictates the requirements for the loop filter and constitutes a major contribution to the in-band phase noise at the ADPLL output. With the assumption of uniform white noise, the in-band phase noise contribution from the FDC can be estimated as follows [111]

$$\mathcal{L} \cong \frac{\pi^2}{3} \cdot \frac{(f_{\mathrm{out}} \cdot T_{\mathrm{res}})^2}{f_{\mathrm{ref}}}, \tag{7.5}$$

which amounts to -87.6 dBc/Hz for the same design parameters given above; this is adequately low for the target application.

7.2.2.2 Adaptive frequency calibration

The adaptive frequency calibration (AFC) is essential for fast settling of the frequency synthesizer with a wide tuning range. It has two operation phases corresponding to two DCO control words, *DAc* and *DAf*. The first phase, with *DAc* being active, is characterized by frequency tuning steps that are significantly larger than the FDC resolution. A modified binary search algorithm [68] is employed for this phase, which allows for its completion within about 10 reference clock cycles. Fig. 7.18(a) depicts the logic branch for this phase. The frequency error is compared with a predefined threshold value, and a finite state machine (FSM) proceeds with a binary search based on the comparison result. This first phase finishes when the phase error magnitude goes below the threshold value or the step size of

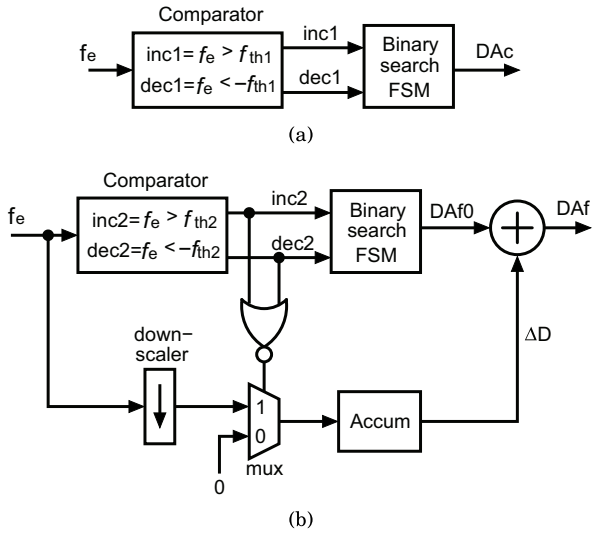


Figure 7.18. AFC logic (a) for the DAc branch and (b) the DAf branch.

the binary search is reduced to zero. During the second phase, the control word *DAf* is active, which is characterized by frequency steps close to the estimated FDC resolution. For reliable frequency calibration in the presence of FDC non-idealities, the second phase uses a modified logic, as shown in Fig. 7.18(b). A binary search is executed when the frequency error is significantly larger than the estimated FDC resolution, while further fine adjustment is made to *DAf* after frequency error approaches the level of FDC resolution. The second phase finishes, when output, ΔD, from the fine adjustment branch changes its sign or remains constant for a specified amount of time. The whole AFC process can finish in about 30 reference clock cycles, with a worst-case calibration accuracy of about 4 MHz.

7.2.2.3 Digitally controlled oscillator core

The oscillation frequency of a basic ring oscillator, with a simplifying assumption of equal delay for each stage, can be expressed with a well-known formula,

$$f_{\mathrm{osc}} = \frac{1}{2N \cdot \tau} = \frac{I_{\mathrm{DD}}}{2N \cdot V_{\mathrm{osc}} \cdot C_L}, \tag{7.6}$$

where N is the number of stages and V_{osc} the voltage amplitude, while τ, I_{DD} and C_L are, respectively, the delay, charging/discharging current and capacitance load for each stage. Frequency tuning of a ring oscillator can be done through both the current, I_{DD}, and capacitance, C_L. Compared with an LC oscillator, a ring oscillator can achieve a relatively large

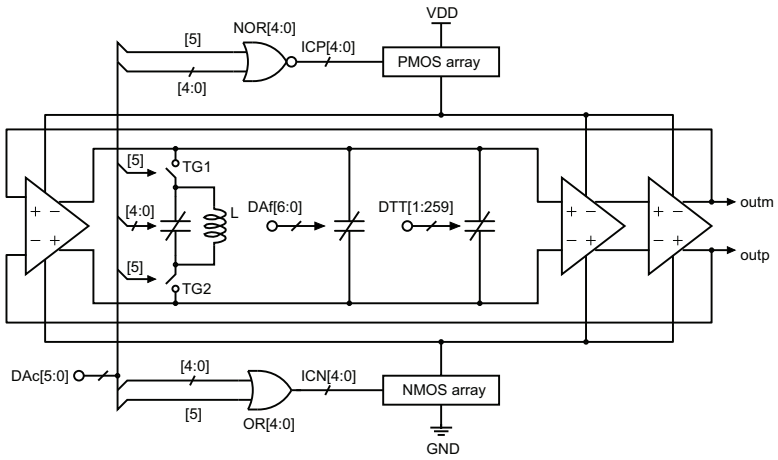


Figure 7.19. Schematic of the DCO core.

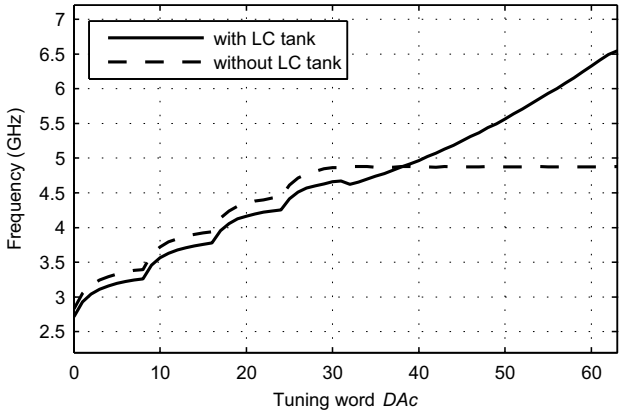


Figure 7.20. Simulated coarse frequency tuning of the implemented ring oscillator as compared with one without the LC tank.

frequency tuning ratio, but its oscillation is limited to lower frequencies. Parasitic capacitances arising from devices and wiring prevent the arbitrary reduction of the load capacitance, C_L. A large current, I_{DD}, needs large inverters, which introduce more parasitic capacitance and can eventually prevent oscillation frequency from further increasing. In addition, power consumption will be an issue when the current, I_{DD}, becomes too large.

To overcome the above issues and achieve a wide tuning range at high frequencies with reduced power consumption, we introduced an LC tank to a ring oscillator. The schematic of the DCO core is shown in Fig. 7.19. It is a three-stage ring oscillator except for the LC tank at the output of the first stage. The basic idea behind using an LC tank is to provide inductive impedance to offset the parasitic capacitances and thus raise

the oscillation frequency. The connection for the LC tank is controlled by the *DAc*, the tuning word dedicated to coarse frequency calibration. With the LC tank switched off from the signal path, the ring oscillator operates as a conventional oscillator, where the transistors in the PMOS and NMOS arrays can be switched to control the current and coarsely tune the oscillation frequency over a certain range. When the LC tank is connected, the bias current is set to be constant. With the impedance of the LC tank designed to be inductive over the target frequency range, the connection of the LC tank partially offsets the load capacitance and, as a result, raises the oscillation frequency to a higher range. Meanwhile, the *DAc* can switch individual cells in the capacitor bank of the LC tank, changing the tank impedance to coarsely tune the frequency. Since the LC tank raises the oscillation frequency by reducing the load capacitance instead of increasing the current, low power consumption can be achieved. Fig. 7.20 shows the simulated coarse frequency tuning for the ring oscillator compared with otherwise the same oscillator but without the LC tank and the switches (TG1 and TG2). It can be observed that using the LC tank extends the overall frequency tuning range to a substantially higher frequency. The parasitic capacitance introduced by the switches only slightly lowers the oscillation frequency when they are open to disconnect the LC tank.

The 1-nH inductor is used merely to cancel the load capacitance. Its quality factor is not as important as it is for a typical LC oscillator, and a low-Q miniaturized inductor would be sufficient. We chose to use the minimum size available from the technology library, which still has a high quality factor. The coil size could be further reduced if a custom design is adopted to lower the quality factor for area reduction. The binary word, *DAf*, is for further frequency calibration, whereas the thermometer code, *DTT*, is intended for final frequency settling and tracking. The capacitor banks connected to these two control words are realized using NMOS varactors.

7.2.3 Experimental results

The frequency synthesizer is implemented in a 65-nm CMOS process. The physical layout is shown in Fig. 7.21. The chip has an active area of 0.3 mm^2. The measured frequency range was from 2.7 to 6.1 GHz. Fig. 7.22 shows the output frequency measured as a function of the *FCW* value. The perfect linear relationship indicates the ADPLL locks correctly over

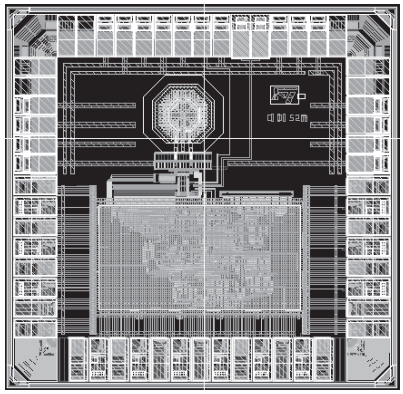


Figure 7.21. Physical layout.

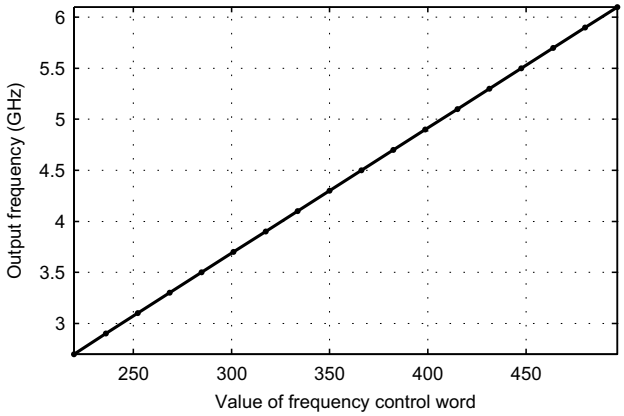


Figure 7.22. Measured output frequency versus FCW value.

the target frequency range. Fig. 7.23 gives the measured phase noise at a 1-MHz offset for different output frequencies. The figure shows that the measured phase noise is better than -85 dBc/Hz over the whole frequency range. Over the frequency range below 4.5 GHz, where the LC tank is switched off, the phase noise degrades as the output frequency increases. Throughout the higher frequency range, the phase noise was slightly better than in the lower frequency range, which was due to the filtering effect of the resonator created from the inductor along with all the capacitor banks and parasitic capacitors at the output of the first oscillator stage. The phase noise performance at an output frequency of 5.5-GHz is shown in Fig. 7.24. It is found in simulations that the DCO is the dominant source of the in-band phase noise. The ring DCO has native phase noise level about -40 dBc/Hz at 10-kHz offset for the given output frequency. With less than 20 dB suppression from the loop, the closed-loop phase noise is expected to be slightly higher than -60 dBc/Hz at 10 kHz

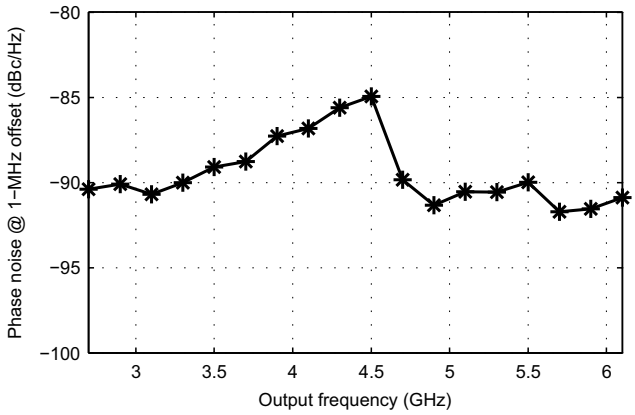


Figure 7.23. Measured phase noise @ 1-MHz offset versus output frequency.

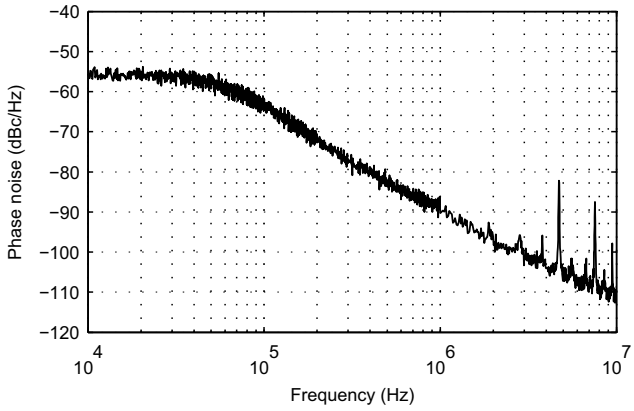


Figure 7.24. Measured phase noise at 5.5 GHz.

from the DCO contribution. Meanwhile, the high-offset DCO phase noise can be folded back to the in-band region due to the sub-sampling at reference frequency and further increase the in-band phase noise level. As a result, the overall in-band phase noise level at the ADPLL output as measured is considerably higher than the estimated contribution of TDC quantization noises. The visible spurs can be attributed to the presence of limit cycles related to the FDC quantization errors. Fig. 7.25 shows the measured spectrum over a 2-GHz span for the same output frequency. The total power consumption varies from 14 mW to 22 mW for a 1.2-V power supply, with about 8 to 16 mW consumed by the oscillator core and around 6 mW consumed by the rest of the circuits. Table 7.5 summarizes the performance of the implemented frequency synthesizer.

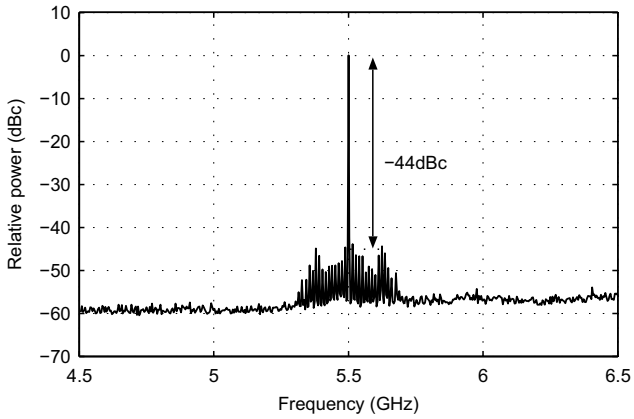


Figure 7.25. Measured output spectrum at 5.5 GHz.

Table 7.5. Performance Summary

Frequency range	2.7–6.1 GHz
Frequency resolution	1 Hz
Phase noise @ 1 MHz	−92 dBc/Hz at 6 GHz
In-band phase noise	−55 dBc/Hz
Reference spur	−44 dBc
Power consumption	14 – 22 mW
Settling time	< 20 μS
Active area	0.3 mm^2
Reference frequency	12.3 MHz
Supply voltage	1.2 V
Technology	65-nm 1P6M CMOS

7.3 A 3-7GHz wideband ADPLL frequency synthesizer

In this section, we present an ADPLL that features the separate operations for the VPA and TDC. The VPA is used exclusively for phase digitization during the coarse tuning. After coarse tuning, phase digitization is performed only by the TDC. This separation allows the outputs from the two circuits to be used independently instead of in combination. It is thus free of all the issues associated with the joint operation of the two circuits. Meanwhile, the ADPLL employs an optimized binary search technique for adaptive frequency calibration (AFC). The technique allows a minimum number of reference cycles to be used during each step in the binary search, resulting in remarkably faster frequency settling than before. This section is organized as follows. Subsection 7.3.1 describes the top-level architecture of the ADPLL. Subsection 7.3.2.1 outlines the

principle of the fast AFC technique and its circuit implementation. Experimental results are provided in subsection 7.3.3.

7.3.1 The proposed ADPLL architecture

Fig. 7.26 shows a simplified block diagram of the proposed ADPLL. The VPA and TDC are separated into coarse-tuning and fine-tuning loops built around the digitally controlled oscillator (DCO). The VPA-based coarse-tuning loop performs coarse AFC, whereas the TDC-based fine-tuning loop is for both fine AFC and normal phase locking. The operation is illustrated with a flowchart in Fig. 7.27. Upon due initialization, the ADPLL frequency settling process starts with the coarse AFC, which is followed by the fine AFC stage before proceeding to the phase-locking operation. The progression between different operation modes goes as follows. The ADPLL is initialized with the *reset* signal as an input. De-asserting the *reset* signal activates the coarse-tuning loop for coarse AFC, during which time the fine-tuning loop remains inactive to maintain its initial state. The completion of AFC is detected in its core logic; at this point, the core logic for coarse AFC asserts a *done* signal and freezes its output. The assertion of the *done* signal from the coarse AFC logic activates the fine-tuning loop for fine AFC, with the whole coarse-tuning loop deactivated. During the fine AFC stage, the digital loop filter (DLPF) branch also remains inactive. When the fine AFC stage is finished, its core logic raises its own *done* signal and deactivates itself with its output frozen. The assertion of the *done* signal from the fine AFC logic activates the digital loop filter (DLPF) branch for phase locking.

In the coarse-tuning loop, phase digitization is performed solely by the VPA. The absence of a TDC in this loop is based on the observation that a TDC, if used, could only play an ineffective role in the coarse-tuning mode. As will be demonstrated in the next section, the initial frequency calibration performed by this loop only needs to ensure that the output frequency error is smaller than one half of the reference frequency with a small margin, for the subsequent fine-tuning loop to operate correctly. This large frequency tolerance, exploited by the AFC technique, allows the initial frequency calibration to be finished within a negligibly short amount of time despite the coarse resolution of a VPA. It also obviates the need for most of the fractional bits in the overall *FCW*. As a result, the input *FCW(H)* comprises only the *FCW* integer part and two or three

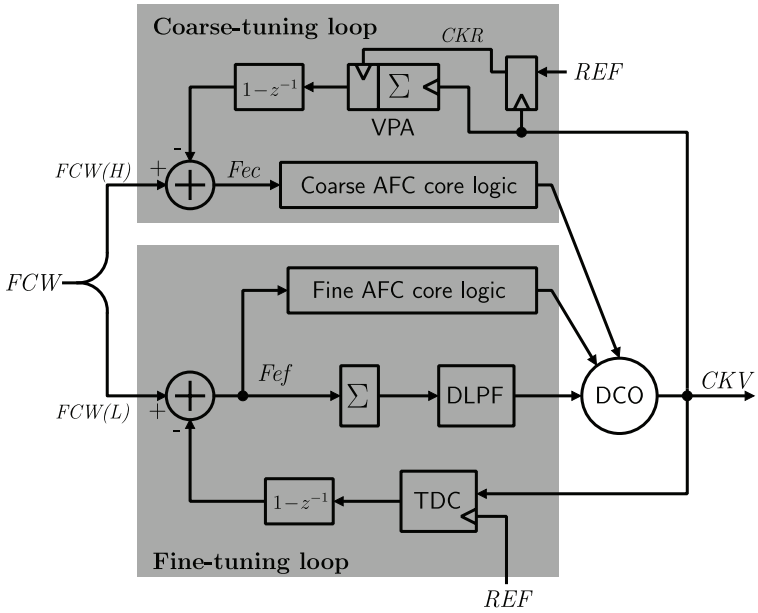


Figure 7.26. The proposed ADPLL architecture.

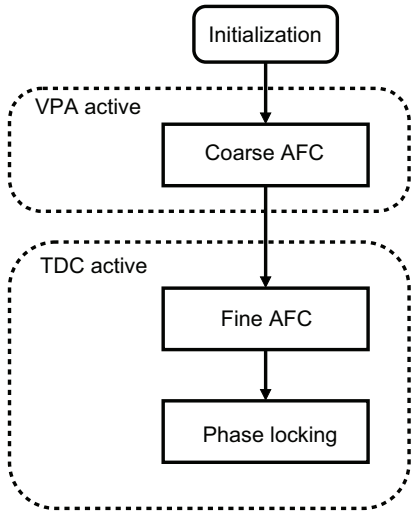


Figure 7.27. Progression flowchart of the proposed ADPLL.

MSBs of the fractional part just to ensure that the truncation error is smaller than the frequency tolerance.

In the fine AFC and phase-locking operation modes, phase digitization is performed exclusively by the TDC, which covers a small range of a single RF cycle. The dramatically reduced range compared with that of a VPA is sufficient because of the relatively small frequency errors after the coarse AFC. As discussed in Chapter 6, the range has no effect as long as the

output frequency error is kept within roughly one half of the reference frequency. Meanwhile, the AFC logic in this loop exploits the fine resolution of the TDC to achieve a fine and fast final frequency calibration. It provides such a fine calibration resolution that the output frequency can immediately be used by the upper-level system after its completion. In other words, no additional frequency settling is required after the loop is closed with the DLPF becoming active. The exclusion of VPA from this loop eliminates the need for clock retiming and the frequency reference can be used directly as the clock signal for the digital logic in fine tuning. Meanwhile, the input *FCW(L)* comprises only the *FCW* fractional part.

With the separate operation of the VPA and TDC in different operation modes, the modified ADPLL architecture is free of all those issues arising from the joint operation of the two circuits. Moreover, the separation of the two loops makes it possible to switch off almost the entire coarse-tuning loop during normal operation of the ADPLL, thus saving the associated power consumption of the loop. With the use of two loops, the word length for most of the arithmetic logic is also dramatically reduced, which allows for a reduction in both the silicon area and the power consumption despite the duplication of some simple digital blocks. Meanwhile, the two-stage arrangement of coarse and fine frequency calibration along with an innovative AFC technique optimally exploits the different resolutions of the VPA and TDC, minimizing the frequency settling time.

7.3.2 Circuit implementation

This subsection discusses the major implementation aspects of the wide-band frequency synthesizer. The emphasis is is again on the functional blocks that are different from the previous designs.

7.3.2.1 Automatic frequency calibration

Fig. 7.28 shows a hardware realization of the proposed binary search technique in the AFC core logic blocks. It can roughly be divided into four functional blocks. The common functionality of a binary search is performed mainly in the step value generation and step execution blocks. The extended frequency comparison block, in conjunction with the end detection block, is largely responsible for the unique functionality required by the proposed approach.

The circuit operation is illustrated with the flow chart shown in Fig. 7.29.

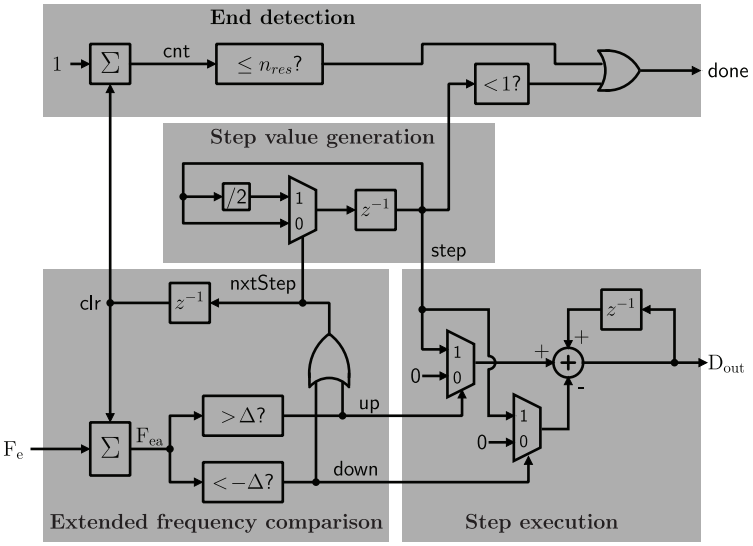


Figure 7.28. The proposed binary search technique in the AFC core logic blocks.

The relevant signals are initialized accordingly during the reset state of the ADPLL. For each reference cycle, the frequency error, F_e, is accumulated and the accumulator output, F_{ea}, is compared with the predefined values, $\pm\Delta$, that correspond to the estimated peak-to-peak magnitude of the phase quantization error.

The comparison result can yield three different scenarios:

- In the first scenario, we have $F_{\rm ea} > \Delta$, which suggests that the output frequency is lower than the target frequency. As a result, the signal *up* is raised to direct the step execution block to step up the DCO frequency.
- In the second scenario, we have $F_{\rm ea} < -\Delta$, which suggests that the output frequency is higher than the target frequency. As a result, the *down* signal is raised to step down the DCO frequency.
- The third scenario arises when the accumulated frequency error becomes small and we have $-\Delta \leq F_{\rm ea} \leq -\Delta$. In this scenario, none of the above signals are asserted and the binary search remains in the current step with the output and step size unchanged.

In both of the first two scenarios, the signal *nxtStep* is asserted, which allows the step value to be halved by a simple bit shifting for the next search step. The content of the frequency error accumulator and the accumulation cycle counter will be cleared in the next reference cycle by the signal *clr*, which is derived from the *nxtStep* signal with a unit delay. In the third scenario, the frequency error is accumulated for each reference

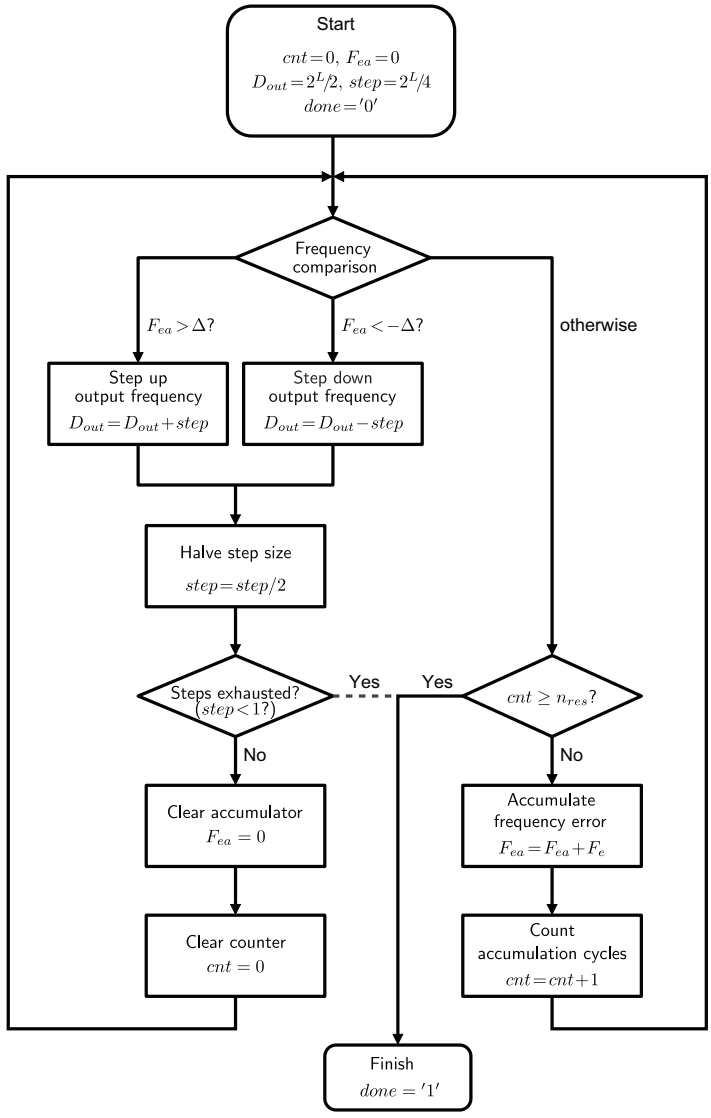


Figure 7.29. A flow chart of the AFC technique (L being the D_{out} word length).

cycle and the number of cycles is counted. If the ADPLL output frequency error is larger than the expected calibration resolution, the circuit will exit the third scenario and enter one of the first two scenarios after a certain number of reference cycles. Otherwise, the counted accumulation number, *cnt*, will reach the predefined maximum value, N_{max}, and terminate the binary search. Upon completion, the *done* signal is raised, which stalls the clock for the AFC circuit.

Note that the binary search will also finish if the last step is reached, which can be detected by monitoring the step value. This, however, means

that the minimum frequency step is too large for the expected calibration resolution, and its occurrence is usually prevented by ensuring in the design that the minimum frequency step is sufficiently small. This exit condition, as indicated by the dashed line in the flow chart, is only for backup purposes. It makes the circuit more robust by avoiding the possibility of indefinite iteration when the minimum step accidentally happens to be too large due to design or implementation issues. In a proper design, the exit condition corresponds to a possible occurrence of the rare retiming metastability, which can be corrected by rerunning the binary search.

Another important consideration is related to the loop latency of a digital PLL. When the DCO output frequency is updated with its control word, its detection comes one reference cycle later since there are two unit delays in the ADPLL loop, one at the DCO input and the other at the phase detection point. In the reference cycle following the frequency word update of the DCO, the detected frequency error is an outdated one. It corresponds to the frequency prior to the updating, and thus should not be used. This is done in the circuit by clearing the accumulator when one step is finished, as shown in both the circuit schematic and flow chart. This leads to an additional delay of one reference cycle introduced to each iteration step. However, it should be noted that the loop latency of two unit delays is the result of digital frequency control, which entails a unit delay at the DCO input to align and synchronize different bits of the frequency control word. It is neither associated with any binary search technique, nor is it unique to the specific ADPLL architecture.

7.3.2.2 Two-stage ring oscillator

The DCO in the design is based on a two-stage ring oscillator as shown in Fig. 7.30. The reduction of one stage as compared with a three-stage counterpart allows for higher oscillation frequencies and a wide tuning range. The transistors in each inverter cell are sized based on simulations to sustain the oscillation condition over a wide frequency range [75].

Frequency control is performed through the current tuning with the two cascaded tail transistor banks. For coarse frequency tuning, an array of 63 NMOS transistors $Mn_a(1, 2, 3...63)$ as controlled through a thermometer code $D_P[1:63]$ are individually sized based on simulations to achieve approximately linear D/F conversion with minimized step variation. As a result, it reduces the required tuning ranges of the finer tuning banks.

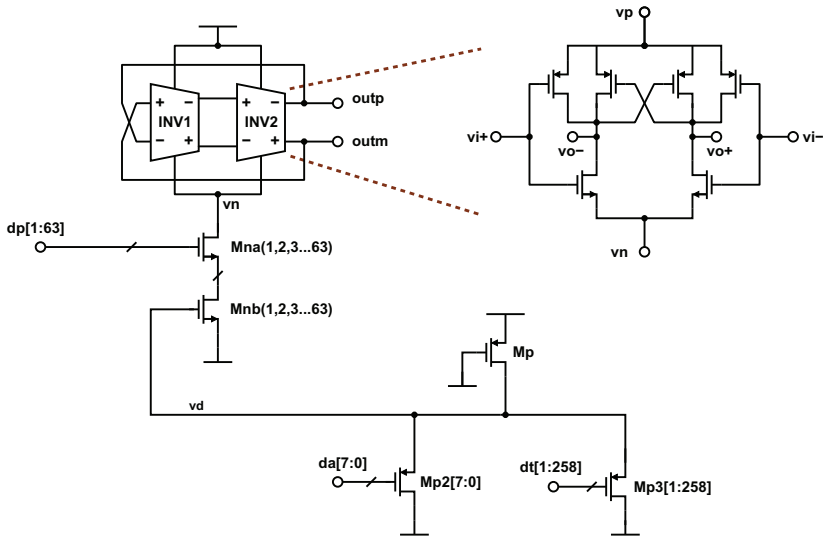


Figure 7.30. The two-stage ring DCO.

The source nodes of the transistors, $Mn_a(1,2,3...63)$, are separate from each other, but each of them is connected to the source of a corresponding transistor in the lower bank, $Mn_b(1,2,3...63)$. Medium and fine frequency-tuning steps are achieved through voltage control of the bottom tail NMOS transistor bank $Mn_b(1,2,3...63)$ with the control voltage generated by a nonlinear D/A converter. In the D/A converter comprises, a PMOS (Mp) on the top and two PMOS banks ($Mp2[7:0]$ and $Mp3[1:258]$) at the bottom constitute a nonlinear voltage divider. The transistors in $Mp2[7:0]$ are binary weighted for a medium-step frequency calibration, while the transistors in $Mp3[1:258]$ is unary weighted for the fine frequency tuning with the thermometer-code control word, $D_t[1:258]$. It should be noted that all the transistors in the D/A converter are P-type. The nonlinearity of the D/A is in the opposite manner as compared with the nonlinearity in the voltage-to-current relation through the NMOS transistors, $Mn_b(1,2,3...63)$. This allows for a good linearity in the overall conversion from the control digital to the current and to the oscillation frequency, which is based on our original work [15].

7.3.3 Experimental results

The ADPLL has been implemented using a 65-nm CMOS with both the architectural modification and the proposed AFC technique. The DCO is based on a ring oscillator to cover a wide frequency range of 3 to 6 GHz. Four current-switching banks are employed in the DCO for frequency tun-

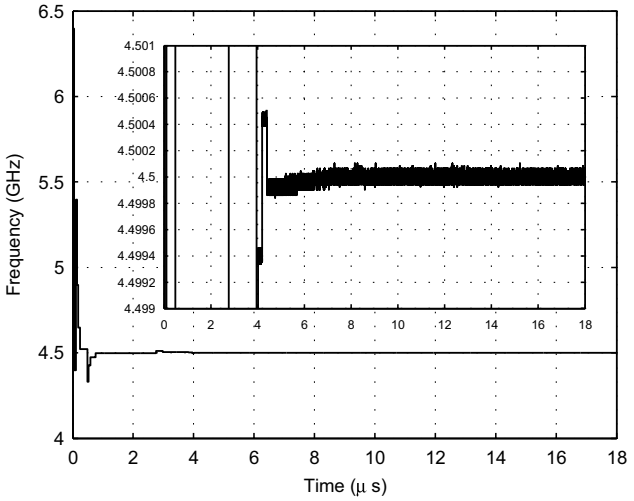


Figure 7.31. Simulated ADPLL frequency settling.

ing to allow for a large tuning range with a fine frequency resolution. Two of the tuning banks are used for initial frequency calibration by the coarse-tuning loop. Correspondingly, there are two binary search stages in the coarse AFC logic, with most of the logic shared using multiplexing. The final ADPLL fine-tuning branch controls the last tuning bank with the finest resolution. Theoretical calculations predict a worst-case settling time of about 6 μs for a ± 200-kHz frequency tolerance. Both simulations and measurements showed that the ADPLL have a maximum settling time that is slightly shorter than the theoretical predication. The result amounts to a reduction of 14 μs and a speed gain of more than three compared with our previous ADPLL design, as described in section 7.2, which was based on the typical architecture with a basic binary search technique. Fig. 7.31 shows the overall frequency settling process of the ADPLL for a target frequency of 4.5-GHz. The measured frequency settling process for the same target frequency is shown in Fig. 7.32. The achieved frequency settling performance is clearly beyond the current state-of-the-art performance. It should be noted that DCO nonlinearity and other non-ideal factors are the reasons for the difference between the simulated and measured frequency settling processes. However, their effect on the overall settling time was found to be quite limited.

A die photograph is shown in Fig. 7.33, where the entire ADPLL occupies an active area of about 0.07 mm^2. Besides the DCO, the VPA and TDC were also custom designed for high-speed operations to handle the RF signal. The rest loop circuitry is realized as a pure digital logic block,

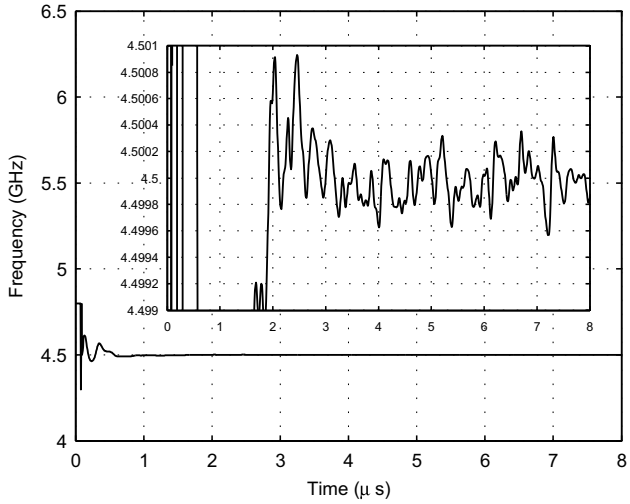


Figure 7.32. Measured frequency settling.

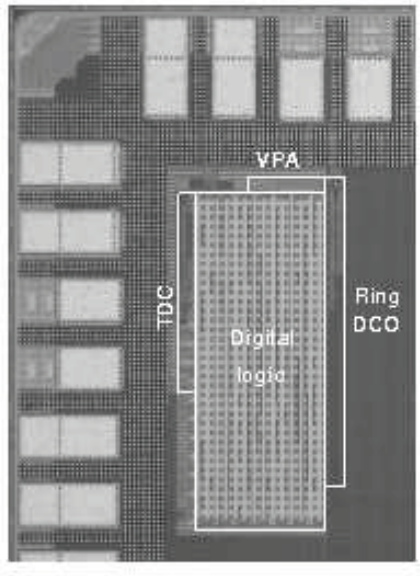


Figure 7.33. Die photograph.

which occupies approximately 0.05 mm^2. This digital area is about one-third of the digital area in our previous ADPLL design, as described in section 7.2. The substantial reduction can mainly be attributed to the architectural modification associated with the improved phase digitization approach.

The entire ADPLL has a maximum power consumption of about 10 mW from a 1.2-V supply. The measured output frequency range is from 2.7 to 7.3 GHz. The measured phase noise at a 1-MHz frequency offset is

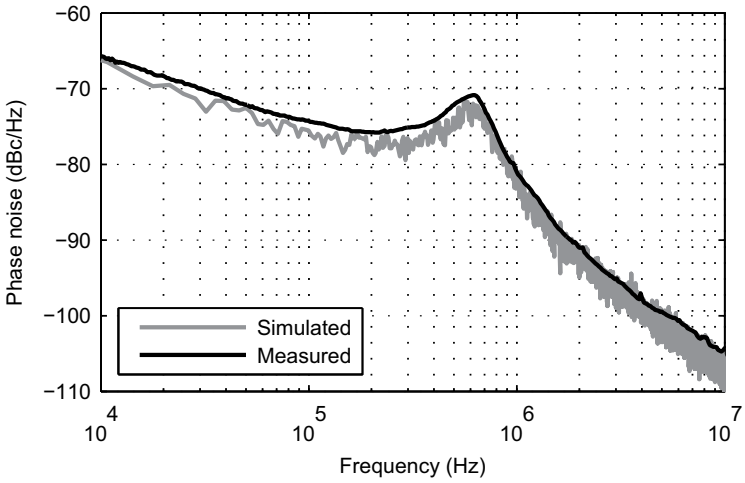


Figure 7.34. Measured and simulated ADPLL phase noise.

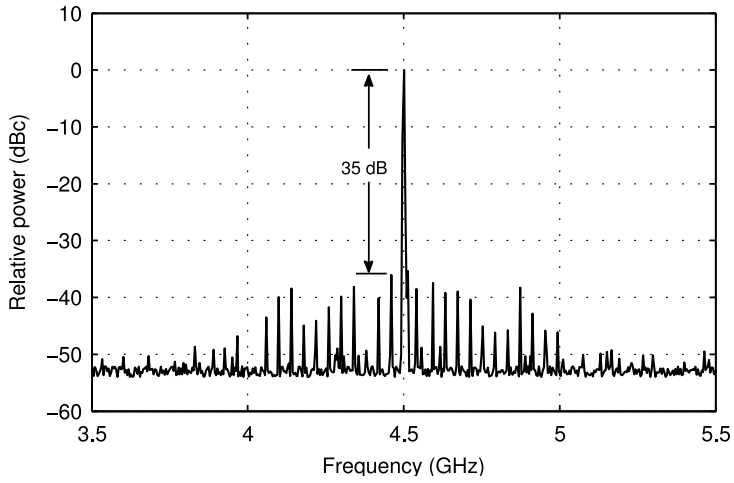


Figure 7.35. Measured ADPLL spectrum (the major spurs appear at intervals of the reference frequency).

about -80 dBc/Hz and the in-band phase noise level is about -70 dBc/Hz. Fig. 7.34 shows an example of measured phase noise at a 4.5-GHz output frequency along with the simulated result. It should be noted that the noisy ring DCO also limits the performance of the ADPLL in-band phase noise. This is because the excessive in-band phase noises of the DCO can still be dominating, even after they have been greatly suppressed by the feedback loop. Meanwhile, the far-off phase noises of the DCO are folded back to low offset frequencies due to the subsampling effect of the feedback path [55]. The measured spectrum is shown in Fig. 7.35. The relatively high reference spurs as observed in the spectrum are associ-

Table 7.6. Performance summary

Output frequency range	2.7–7.3 GHz
Loop bandwidth	800 kHz
Power consumption	10 mW
Phase noise @ 1 MHz	−80 dBc/Hz
In-band phase noise	−70 dBc/Hz
Settling time	6 μs
Supply voltage	1.2 V
Reference frequency	40 MHz
Active area	0.07 mm^2
Technology	65-nm 1P6M CMOS

ated with the use of a digital loop filter and the ring oscillator. A ring oscillator, as adopted in our implementation, is sensitive to supply and ground disturbance, which effectively picks up the interferences from digital switching in the reference clock domain and generates reference spurs in the output spectrum. Meanwhile, a digital loop filter does not suppress reference spurs; instead it contributes to them due to its clocking at the reference frequency.

Table 7.6 provides a performance summary of the implemented ADPLL, where the values refer to the worst-case scenarios. The overall achieved result is remarkably favorable compared with existing ADPLL designs for a wideband application. It fully demonstrates the advantages of the modified ADPLL architecture and the frequency calibration technique. A performance comparison with prior studies on fast PLL frequency settling is provided in Table 7.7.

7.4 Discussion and summary

In this chapter, three ADPLL frequency synthesizer prototype designs have been presented together with the measurement results. In these prototype designs, several novel circuit techniques as well as an architectural improvement have been proposed and validated. These circuit techniques include an effective TDC power-gating technique, a high-speed VPA topology, a ring oscillator band-extension technique and an optimized

AFC technique based on a dynamic binary search. In the architectural improvement, we were the first to demonstrate that the TDC alone would be adequate for DCO phase digitization in the fine tuning mode, thus sparing us the rather futile attempts of trying to combine a VPA and TDC for a simultaneous operation. We proved that a low-power, low-cost ADPLL-based frequency synthesizer design is made feasible for high-performance wideband wireless applications with the proposed circuit techniques and architectural improvement.

Table 7.7. Comparison with relevant prior arts

Parameter		[112] ASSC'07	[113] JSSC'07	[61] CASII'07	[114] CASI'10	[62] JSSC'10	[115] JSSC'10	[116] MTT'11	[117] CASII'11	**This work**
Operating freq (GHz)		0.4–1.0	8.67–10.12	–	2.3–2.9	9.75–10.17	5.27-5.6	0.4-0.41	39–42	**2.7–7.3**
Freq.	accuracy(kHz)	–	–	–	–	±200	–	–	–	**±200**
settling	time(μs)	2.2	7	50	20	6.9	20	110	15	**6**
In-band phase noise (dBc/Hz)		-100	N/A	-81	-85	-78	-72	-65	-70	**-70**
Phase noise @1MHz (dBc/Hz)		-119	-102	-117	-113	-100	-114.3	-92	-83.9	**-80**
Reference freq. (MHz)		20	40	26	19.2	40	10	10	156.2	**40**
Reference spur (dBc)		-48	-48	-92	-63	-72	-70	-55	-48	**-35**
Power consumption (mW)		22	44	43	19	7.1	19.8	1.26	46	**10**
Active area (mm^2)		0.63	1.35 †	1.5	2.1 †	0.352	1.61	0.14	0.3	**0.07**
Process (nm)		130	180	90	130	90	180	180	90	**65**

† Full chip size including pads.

8. Conclusions

The RFIC design for wireless systems has increasingly embraced digital-intensive approaches to better leverage the down-scaling of CMOS technology. A milestone development in such a trend is the advent of an ADPLL architecture for frequency synthesis [4]. An ADPLL-based frequency synthesizer features a digital-intensive implementation and thus it has better compatibility with nowadays nanoscale CMOS technology, as compared with traditional analog-intensive designs. Meanwhile, an ADPLL-based frequency synthesizer still consumes a significant amount of current and occupies a significant silicon area. In particular, the initial ADPLL architecture features a relatively high-level of complexity and presents plenty of new design challenges.

In this work, the design of ADPLL-based frequency synthesizers is studied. Several techniques for power and cost reduction as well as for performance improvement have been proposed and verified with prototype designs. These circuit techniques include an effective TDC power-gating technique, a high-speed topology of a VPA for integer phase digitization, a ring oscillator band-extension technique, and an optimized AFC technique based on a dynamic binary search. The work demonstrates how the utilization a frequency divider prior to phase digitization in the feedback path makes it possible to apply the ADPLL for wideband frequency synthesis. The author pointed out and proved that the TDC alone is adequate for DCO phase digitization in the fine tuning mode, thus sparing us us the rather futile attempts of trying to combine a VPA and a TDC for simultaneous phase digitization operations, as in other published works in the existing literature.

The proposed circuit techniques and architecture improvements are verified, respectively, with three prototype designs. The first prototype de-

sign is a 2.4-GHz ADPLL, where a high-speed topology was used for the variable phase accumulator to allow for reliable handling of the high-frequency signal from the RF output and a delay-based technique is employed to effectively reduce the TDC power dissipation. The ADPLL features a wide frequency range and good phase noise performance with low power consumption and a small die area. The second prototype design is a wideband frequency synthesizer based on an ADPLL architecture. An LC tank was introduced to extend the frequency tuning range of a ring oscillator. A dual-modulus frequency divider was used in the feedback path to ensure a reduced frequency range so that it can be reliably handled by the VPA and TDC with a cost-effective implementation. For frequency acquisition, a binary search technique was employed for frequency calibration. The third prototype is another wideband ADPLL frequency synthesizer. It features separate use of VPA and TDC in the feedback path for phase digitization as opposed to commonplace joint use for the two circuit blocks. The separation simplifies the implementation, resulting in a reduced silicon area and lower power consumption, while it also allows for more robust operation of the ADPLL. In addition, the ADPLL also employs a new binary search technique for frequency calibration. By allowing a minimum number of reference cycles to be used for each search step, based on the target calibration resolution, the technique is capable of achieving a fine frequency calibration resolution with a greatly reduced time as compared with a conventional binary search technique.

The major design challenges with an ADPLL frequency synthesizer have been with the TDC and the DCO, which are the two major blocks that still involve custom design, while the rest of the loop is essentially fully digital. While the digital frequency control used with a DCO overcomes the issues associated with an analog voltage control, it introduces frequency quantization that tends to degrade the purity of the output spectrum during active operation. In a wireless application with stringent specifications on spectrum purity, the DCO needs to be designed with a fine frequency resolution and good linearity in the tracking mode. The use of multiple tuning banks is usually a necessity to ease the implementation by allowing the finest resolution to be used only for frequency tracking and the overall tuning range to be covered with coarse steps. With the need for the finest resolution and for covering a sufficient range to accommodate frequency change due to temperature and supply variations, implementing the fine-tuning bank could remain challenging and prohibitive. An

effective technique for easing the implementation is to dither the unit capacitors with digital SDM to create an effective fractional frequency resolution. It does not only overcome possible limitations with the minimum device dimensions but also greatly reduces the required number of cells. The TDC that is used for phase quantization in the feedback path needs to provide a fine quantization resolution with good linearity to minimize its contribution to the overall phase noise and spurious tones. Meanwhile, it needs to be capable of high-speed operation to handle the RF signal from the DCO. With high-speed operation, current consumption becomes an important consideration for the TDC. An effective power-gating technique is usually necessary to keep the average TDC current consumption low, while its impact on the TDC performance metrics, such as linearity, needs to be accounted for.

The techniques covered in this work and others have successfully addressed most of the design challenges. However, new architectural or circuit-level innovations are expected to further ease the design challenges and bring about performance improvements as well as to reduce current consumption. The focus of future work on an ADPLL for frequency synthesis is expected to be on digital or digital-intensive techniques in order to be effective and maintain good compatibility with technology scaling.

Bibliography

[1] A.-J. Annema, B. Nauta, R. van Langevelde, and H. Tuinhout, "Analog circuits in ultra-deep-submicron CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 132–143, Jan. 2005.

[2] L. Lewyn, T. Ytterdal, C. Wulff, and K. Martin, "Analog circuit design in nanoscale CMOS technologies," *Proc. IEEE*, vol. 97, no. 10, pp. 1687–1714, Oct. 2009.

[3] R. B. Staszewski, D. Leipold, K. Muhammad, and P. T. Balsara, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS process," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 11, pp. 815–828, Nov. 2003.

[4] R. B. Staszewski, J. L. Wallberg, S. Rezeq, C.-M. Hung, O. E. Eliezer, S. K. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phone," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.

[5] L. Xu and S. Lindfors, "A high-speed variable phase accumulator for an ADPLL architecture," in *Proc. Int. Symp. Circuits Syst.*, 2008, pp. 1736–1739.

[6] L. Xu, S. Lindfors, and J. Ryynänen, "An ADPLL-based fast start-up technique for sensor radio frequency synthesizers," in *Proc. Intl. Conf. on Electronics, Circuits and Systems*, Aug. 2008, pp. 388–391.

[7] L. Xu, K. Stadius, T. Rapinoja, and J. Ryynänen, "Agile frequency synthesizer for cognitive radios," in *Proc. European Conf. on Circuit Theory and Design*, 2009, pp. 275 –278.

[8] L. Xu, K. Stadius, and J. Ryynänen, "A low-power wide-band digital frequency synthesizer for cognitive radio sensor units," in *Proc. European Solid-State Circuits Conf.*, 2009, pp. 184–187.

[9] L. Xu, S. Lindfors, K. Stadius, and J. Ryynänen, "A digitally controlled 2.4-GHz oscillator in 65-nm CMOS," *Analog Integrated Circuits and Signal Processing*, vol. 58, no. 1, pp. 35–42, Jan. 2009.

[10] ——, "A 2.4-GHz low-power all-digital phase-locked loop," in *Proc. Custom Integrated Circuits Conf.*, 2009, pp. 331–334.

[11] L. Xu, K. Stadius, and J. Ryynänen, "A wide-band digitally controlled ring oscillator," in *Proc. Int. Symp. Circuits Syst.*, May 2010, pp. 1983–1986.

[12] L. Xu, S. Lindfors, K. Stadius, and J. Ryynänen, "A 2.4-GHz low-power all-digital phase-locked loop," *IEEE J. Solid-State Circuits*, vol. 45, no. 8, pp. 1513–1521, Aug. 2010.

[13] L. Xu, K. Stadius, and J. Ryynänen, "An all-digital PLL frequency synthesizer with an improved phase digitization approach and an optimized frequency calibration technique," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 11, pp. 2481–2494, Nov. 2012.

[14] L. Xu and S. Lindfors, "Phase accumulator for digital phase locked loop," WO 2009053531, Apr. 2009.

[15] L. Xu, K. Stadius, and J. Ryynänen, "Digitally controlled delay," WO 2012156952, Nov. 2012.

[16] A. Abidi, "Direct-conversion radio transceivers for digital communications," *IEEE J. Solid-State Circuits*, vol. 30, no. 12, pp. 1399–1410, Dec. 1995.

[17] W. Namgoong and T. Meng, "Direct-conversion RF receiver design," *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 518–529, Mar. 2001.

[18] P. Zhang, L. Der, D. Guo, I. Sever, T. Bourdi, C. Lam, A. Zolfaghari, J. Chen, D. Gambetta, B. Cheng, S. Gowder, S. Hart, L. Huynh, T. Nguyen, and B. Razavi, "A single-chip dual-band direct-conversion IEEE 802.11a/b/g WLAN transceiver in 0.18-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1932–1939, Sept. 2005.

[19] N. Poobuapheun, W.-H. Chen, Z. Boos, and A. Niknejad, "A 1.5-V 0.7–2.5-GHz CMOS quadrature demodulator for multiband direct-conversion receivers," *IEEE J. Solid-State Circuits*, vol. 42, no. 8, pp. 1669–1677, Aug. 2007.

[20] B. Razavi, *RF Microelectronics*. New Jersey, USA: Prentice-Hall PTR, 1998.

[21] P.-I. Mak, S.-P. U, and R. Martins, "Transceiver architecture selection: Review, state-of-the-art survey and case study," *IEEE Circuits and Systems Magazine*, vol. 7, no. 2, pp. 6–25, 2007.

[22] Q. Gu, *RF System Design of Transceivers for Wireless Communications*. USA: Springer, 2006.

[23] A. Luzzatto and G. Shirazi, *Wireless Transceiver Design: Mastering the Design of Modern Wireless Equipment and Systems*. England: John Wiley & Sons Ltd., 2007.

[24] A. Hajimiri and T. Lee, "A general theory of phase noise in electrical oscillators," *IEEE J. Solid-State Circuits*, vol. 33, no. 2, pp. 179–194, Feb. 1998.

[25] D. Banerjee, *PLL Performance, Simulation, and Design*, 4th ed., 2006, ch. 15, pp. 114–124.

[26] K. Stadius, T. Rapinoja, J. Kaukovuori, J. Ryynänen, and K. Halonen,

"Multitone fast frequency-hopping synthesizer for UWB radio," *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 8, pp. 1633–1641, Aug. 2007.

[27] A. Marques, M. Steyaert, and W. Sansen, "Theory of PLL fractional-N frequency synthesizers," *Wireless Networks*, vol. 4, no. 1, pp. 79–85, Jan. 1998.

[28] W. F. Egan, *Frequency Synthesis by Phase Lock*, 2nd ed. New York.: John Wiley & Sons, 1999.

[29] R. Ahola, "Integrated radio frequency synthesizers for wireless applications," Ph.D. dissertation, Helsinki University of Technology, Espoo, Finland, 2000.

[30] H. C. Luong and G. C. T. Leung, *Low-Voltage CMOS RF Frequency Synthesizers*. United Kingdom: Cambridge University Press, 2004.

[31] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*. Boston, USA: Springer Science + Business Media, Inc., 2005.

[32] A. Torosyan and J. Willson, A.N., "Analysis of the output spectrum for direct digital frequency synthesizers in the presence of phase truncation and finite arithmetic precision," in *Proc. Int. Symp. Image and Signal Processing and Analysis*, 2001, pp. 458–463.

[33] Y.-D. Wu, C.-M. Lai, C.-C. Lee, and P.-C. Huang, "A quantization error minimization method using DDS-DAC for wideband fractional-N frequency synthesizer," *IEEE J. Solid-State Circuits*, vol. 45, no. 11, pp. 2283–2291, 2010.

[34] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553–559, May 1993.

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

[36] B. D. Muer and M. Steyaert, "On the analysis of $\Delta\Sigma$ fractional-N frequency synthesizers for high-spectral purity," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 11, pp. 784–793, Nov. 2003.

[37] S. Meninger and M. Perrott, "A fractional-N frequency synthesizer architecture utilizing a mismatch compensated PFD/DAC structure for reduced quantization-induced phase noise," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 11, pp. 839–849, Nov. 2003.

[38] M. Kozak and I. Kale, "Rigorous analysis of delta-sigma modulators for fractional-N PLL frequency synthesis," *IEEE Trans. Circuits Syst. I*, Jun. 2004.

[39] P.-E. Su and S. Pamarti, "Fractional-N phase-locked-loop-based frequency synthesis: A tutorial," *IEEE Trans. Circuits Syst. II*, vol. 56, no. 12, pp. 881–885, Dec. 2009.

[40] K. Shu and E. Sanchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*. Springer, 2005.

[41] G. Chien and P. R. Gray, "A 900-MHz local oscillator using a DLL-based

frequency multiplier technique for PCS applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 12, pp. 1996–1999, Dec. 2000.

[42] D. Foley and M. Flynn, "CMOS DLL-based 2-V 3.2-ps jitter 1-GHz clock synthesizer and temperature-compensated tunable oscillator," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, pp. 417 –423, Mar. 2001.

[43] S. Ye and I. Galton, "Techniques for phase noise suppression in recirculating DLL," *IEEE J. Solid-State Circuits*, vol. 39, no. 8, pp. 1222–1230, Aug. 2004.

[44] T.-C. Lee and K.-J. Hsiao, "The design and analysis of a DLL-based frequency synthesizer for UWB application," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1245–1252, Jun. 2006.

[45] Q. Du, J. Zhuang, and T. Kwasniewski, "A low-phase noise, anti-harmonic programmable DLL frequency multiplier with period error compensation for spur redution," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 11, pp. 1205–1209, Nov. 2006.

[46] T. Rapinoja, K. Stadius, L. Xu, S. Lindfors, R. Kaunisto, A. Parssinen, and J. Ryynänen, "A digital frequency synthesizer for cognitive radio spectrum sensing applications," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 5, pp. 1339–1348, May 2010.

[47] O. Casha, I. Grech, F. Badets, D. Morche, and J. Micallef, "Analysis of the spur characteristics of edge-combining DLL-based frequency multipliers," *IEEE Trans. Circuits Syst. II*, vol. 56, no. 2, pp. 132–136, Feb. 2009.

[48] P. Torkzadeh, A. Tajalli, and M. Atarodi, "A wide tuning range, 1 GHz-2.5 GHz DLL-based fractional frequency synthesizer," in *Proc. Int. Symp. Circuits Syst.*, May 2005, pp. 5031–5034.

[49] R. B. Staszewski and P. T. Balsara, "Phase-domain all-digital phase-locked loop," *IEEE Trans. Circuits Syst. II*, vol. 52, no. 3, pp. 159–163, Mar. 2005.

[50] J. Bergmans, "Effect of loop delay on stability of discrete-time PLL," *IEEE Trans. Circuits Syst. I*, vol. 42, no. 4, pp. 229–231, Apr. 1995.

[51] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Hoboken, NJ, USA.: John Wiley & Sons, 2005.

[52] D. Lee, "Analysis of jitter in phase-locked loops," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 11, pp. 704–711, Nov. 2002.

[53] R. B. Staszewski, D. Leipold, C.-M. Hung, and P. T. Balsara, "TDC-based frequency synthesizer for wireless applications," in *Proc. IEEE Radio Freq. Integr. Circuits Symp.*, Jun. 2004, pp. 215–218.

[54] R. B. Staszewski, C.-M. Hung, N. Barton, M.-C. Lee, and D. Leipold, "A digitally controlled oscillator in a 90 nm digital CMOS process for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2203–2211, Nov. 2005.

[55] P. Madoglio, M. Zanuso, S. Levantino, C. Samori, and A. L. Lacaita, "Quantization effects in all-digital phase-locked loops," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 12, pp. 1120–1124, Dec. 2007.

[56] B. Razavi, "Cognitive radio design challenges and techniques," *IEEE J. Solid-State Circuits*, vol. 45, no. 8, pp. 1542–1553, Aug. 2010.

[57] C.-M. Hsu, M. Z. Straayer, and M. H. Perrott, "A low-noise wide-BW 3.6-GHz digital $\Delta\Sigma$ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2776–2786, Dec. 2008.

[58] A. Kral, F. Behbahani, and A. Abidi, "RF-CMOS oscillators with switched tuning," in *Proc. Custom Integrated Circuits Conf.*, may 1998, pp. 555 –558.

[59] R. B. Staszewski, C.-M. Hung, D. Leipold, and P. Balsara, "A first multi-gigahertz digitally controlled oscillator for wireless applications," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 11, pp. 2154–2164, Nov. 2003.

[60] R. B. Staszewski, G. Shriki, and P. T. Balsara, "All-digital PLL with ultra fast acquisition," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2005, pp. 289–292.

[61] R. B. Staszewski and P. T. Balsara, "All-digital PLL with ultra fast settling," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 2, pp. 181–185, Feb. 2007.

[62] S.-Y. Yang, W.-Z. Chen, and T.-Y. Lu, "A 7.1 mW, 10 GHz all digital frequency synthesizer with dynamically reconfigured digital loop filter in 90 nm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 3, pp. 578–586, Mar. 2010.

[63] W. Zhang, X. Dai, J. Jin, and J. Zhou, "A novel fast-settling ADPLL architecture with frequency tuning word presetting and calibration," in *Proc. International Conference on ASIC*, Oct. 2009, pp. 1161–1164.

[64] C.-T. Wu, W.-C. Shen, W. Wang, and A.-Y. Wu, "A two-cycle lock-in time ADPLL design based on a frequency estimation algorithm," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 6, pp. 430–434, Jun. 2010.

[65] X. Chen, J. Yang, and L.-X. Shi, "A fast locking all-digital phase-locked loop via feed-forward compensation technique," *IEEE Trans. VLSI Syst.*, vol. 19, no. 5, pp. 857–868, May 2011.

[66] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 30, no. 4, pp. 412–422, Apr. 1995.

[67] T.-D. Chiueh, J.-B. Yang, and J.-S. Wu, "Design and implementation of a low-voltage fast-switching mixed-signal-controlled frequency synthesizer," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 10, pp. 961–971, Oct. 2001.

[68] H. il Lee, J.-K. Cho, K.-S. Lee, I.-C. Hwang, T.-W. Ahn, K.-S. Nah, and B.-H. Park, "A Σ-Δ fractional-N frequency synthesizer using a wide-band integrated VCO and a fast AFC technique for GSM/GPRS/WCDMA applications," vol. 39, no. 7, pp. 1164–1169, July 2004.

[69] L. Lu, Z. Gong, Y. Liao, H. Min, and Z. Tang, "A 975-to-1960MHz fast-locking fractional-N synthesizer with adaptive bandwidth control and 4/4.5 prescaler for digital TV tuners," in *Proc. IEEE Intl. Solid-State Circuits Conf., Digest of Technical Papers*, 2009.

[70] J. Cranickx and M. Steyaert, "Low-noise voltage-controlled oscillators using enhanced LC-tanks," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 794–804, Dec. 1995.

[71] A. Hajimiri and T. H. Lee, *The Design of Low Noise Oscillators*. Kluwer Academic Publishers, 1999.

[72] A. Rezayee and K. Martin, "A 10-Gb/s clock recovery circuit with linear phase detector and coupled two-stage ring oscillator," in *Proc. European Solid-State Circuits Conf.*, Sept. 2002, pp. 419–422.

[73] E. Hegazi, J. Rael, and A. Abidi, *The Designer's Guide to High-Purity Oscillators*. Kluwer Academic Publishers, 2009.

[74] A. Mazzanti and P. Andreani, "Class-C harmonic CMOS VCOs, with a general result on phase noise," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2716–2729, 2008.

[75] B. Fahs, W. Ali-Ahmad, and P. Gamand, "A two-stage ring oscillator in $0.13 - \mu$m CMOS for UWB impulse radio," *IEEE Trans. Microw. Theory Tech.*, vol. 57, no. 5, pp. 1074–1082, 2009.

[76] E. Rubiola, *Phase Noise and Frequency Stability in Oscillators*. Kluwer Academic Publishers, 2009.

[77] K. Stadius, "Integrated RF oscillators and LO signal generation circuits," Ph.D. dissertation, Aalto University, Espoo Finland, 2010.

[78] P. Hammer and P. Bakken, "2.4 GHz CMOS VCO with multiple tuning inputs," *IEE Electronics Letters*, vol. 38, no. 16, pp. 874–876, Aug. 2002.

[79] K. Manetakis, D. Jessie, and C. Narathong, "A CMOS VCO with 48% tuning range for modern broadband systems," in *Proc. Custom Integrated Circuits Conf.*, Oct. 2004, pp. 265–268.

[80] A. Loke, R. Barnes, T. Wee, M. Oshima, C. Moore, R. Kennedy, and M. Gilsdorf, "A versatile 90-nm CMOS charge-pump PLL for SerDes transmitter clocking," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1894–1907, Aug. 2006.

[81] S. Pamarti and I. Galton, "LSB dithering in MASH delta-sigma D/A converters," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 4, pp. 779–790, Apr. 2007.

[82] K. Hosseini and M. Kennedy, "Maximum sequence length MASH digital delta-sigma modulators," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 12, pp. 2628–2638, Dec. 2007.

[83] Z. Xu, J. G. Lee, and S. Masui, "Self-dithered digital delta-sigma modulators for fractional-N frequency synthesizers," *Technical report of IEICE. ICD*, vol. 110, no. 140, Jul 2010.

[84] R. J. V. D. Plassche, "Dynamic element matching for high-accuracy monolithic D/A converters," *IEEE J. Solid-State Circuits*, vol. SC-11, no. 6, pp. 795–800, Dec. 1976.

[85] R. T. Baird, "Linearity enhancement of multibit $\Delta\Sigma$ A/D and D/A converters using data weighted average," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 12, pp. 753–762, Dec. 1995.

[86] I. Fujimori, L. Longo, A. Hairpetian, K. Seiyama, S. Kosic, J. Cao, and S.-L. Chan, "A 90-dB SNR 2.5-MHz output-rate ADC using cascaded multibit delta-sigma modulation at 8x oversampling ratio," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 1820–1828, Dec. 2000.

[87] I. Galton, "Why dynamic-element-matching DACs work," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 2, pp. 69–74, Feb. 2010.

[88] M. Neitola and T. Rahkonen, "A generalized data-weighted averaging algorithm," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 115–119, Feb. 2010.

[89] E. Temporiti, C. Weltin-Wu, D. Baldi, R. Tonietto, and F. Svelto, "A 3 GHz fractional all-digital PLL with a 1.8 MHz bandwidth implementing spur reduction techniques," *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 824–834, Mar. 2009.

[90] M. Zanuso, S. Levantino, D. Tasca, D. Raiteri, C. Samori, and A. Lacaita, "A glitch-corrector circuit for low-spur ADPLLs," in *Proc. Intl. Conf. on Electronics, Circuits and Systems*, 2009, pp. 595–598.

[91] M. Lee, M. E. Heidari, and A. A. Abidi, "A low-noise wideband digital phase-locked loop based on a coarse-fine time-to-digital converter with subpicosecond resolution," *IEEE J. Solid-State Circuits*, vol. 44, no. 10, pp. 2808–2816, Oct. 2009.

[92] S. Henzler, *Time-to-digital Converter*. Springer, 2010.

[93] A. M. Abas, A. Bystrov, D. Kinniment, O. Maevsky, G. Russell, and A. Yakovlev, "Time difference amplifier," *Electronics Letters*, vol. 38, no. 23, pp. 1437–1438, 2002.

[94] M. Lee and A. A. Abidi, "A 9 b, 1.25 ps resolution coarse-fine time-to-digital converter in 90 nm CMOS that amplifies a time residue," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 769–777, Apr. 2008.

[95] S. Henzler, S. Koeppe, D. Lorenz, W. Kamp, R. Kuenemund, and D. Schmitt-Landsiedel, "A local passive time interpolation concept for variation-tolerant high-resolution time-to-digital conversion," *IEEE J. Solid-State Circuits*, vol. 43, no. 7, pp. 1666–1676, Jul. 2008.

[96] S. Henzler and S. Koeppe, "Apparatus and system with a time delayapparatus and system with a time delay path and method for propagating a timing event path and method for propagating a timing event," U.S. Patent 20100034056, Feb. 2010.

[97] J. Yu, F. F. Dai, and R. Jaeger, "A 12-bit vernier ring time-to-digital converter in 0.13 μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 830–842, Apr. 2010.

[98] B. Helal, M. Straayer, G.-Y. Wei, and M. Perrott, "A low jitter 1.6 GHz multiplying DLL utilizing a scrambling time-to-digital converter and digital correlation," in *IEEE Symp. VLSI Circuits*, June 2007, pp. 166–167.

[99] M. Ercegovac and T. Lang, "Binary counter with counting period of one half adder independent of counter size," *IEEE Trans. Circuits Syst.*, vol. 36, no. 6, pp. 924–926, Jun. 1989.

[100] C.-H. Yeh, B. Parhami, and Y. Wang, "Designs of counters with near minimal counting/sampling period and hardware complexity," in *Proc. Conf. on Signals, Systems and Computers*, 2000, pp. 894–898.

[101] R. B. Staszewski, J. Wallberg, J. Koh, and P. T. Balsara, "High-speed digital circuits for a 2.4-GHz all digital RF frequency synthesizer in 130 nm CMOS," in *Proc. IEEE Dallas/CAS Workshop*, Sept. 2004, pp. 167–170.

[102] R. Staszewski, S. Vemulapalli, P. Vallur, J. Wallberg, and P. Balsara, "1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 220–224, Mar. 2006.

[103] R. B. Staszewski, J. Wallberg, C.-M. Hung, G. Feygin, M. Entezari, and D. Leipold, "LMS-based calibration of an RF digitally controlled oscillator for mobile phones," *IEEE Trans. Circuits Syst. II*, vol. 53, no. 3, pp. 225–229, Mar. 2006.

[104] R. Staszewski, C.-M. Hung, K. Maggio, J. Wallberg, D. Leipold, and P. Balsara, "All-digital phase-domain TX frequency synthesizer for Bluetooth radios in 0.13μm CMOS," in *Proc. IEEE Intl. Solid-State Circuits Conf., Digest of Technical Papers*, vol. 1, Feb. 2004, pp. 272–527.

[105] S. Pamarti, L. Jansson, and I. Galton, "A wideband 2.4-GHz delta-sigma fractional-N PLL with 1-Mb/s in-loop modulation," *IEEE J. Solid-State Circuits*, vol. 39, no. 1, pp. 49–62, Jan. 2004.

[106] M. Gupta and B.-S. Song, "A 1.8-GHz spur-cancelled fractional-N frequency synthesizer with LMS-based DAC gain calibration," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2842–2851, Dec. 2006.

[107] A. Swaminathan, K. J. Wang, and I. Galton, "A wide-bandwidth 2.4 GHz ISM band fractional-N PLL with adaptive phase noise cancellation," *IEEE J. Solid-State Circuits*, vol. 42, no. 12, pp. 2639–2650, Dec. 2007.

[108] C.-L. Ti, Y.-H. Liu, and T.-H. Lin, "A 2.4-GHz fractional-N PLL with a PFD/CP linearization and an improved CP circuit," in *Proc. Int. Symp. Circuits Syst.*, May 2008, pp. 1728–1731.

[109] M. A. Ferriss and M. P. Flynn, "A 14 mW fractional-N PLL modulator with a digital phase detector and frequency switching scheme," *IEEE J. Solid-State Circuits*, vol. 43, no. 11, pp. 2464–2471, Nov. 2008.

[110] K. Wang, A. Swaminathan, and I. Galton, "Spurious tone suppression techniques applied to a wide-bandwidth 2.4 GHz fractional-N PLL," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2787–2797, Dec. 2008.

[111] R. Staszewski, D. Leipold, C.-M. Hung, and P. Balsara, "TDC-based frequency synthesizer for wireless applications," in *Proc. IEEE Radio Freq. Integr. Circuits Symp.*, June 2004, pp. 215–218.

[112] J. Lee, K. Kim, J. Lee, T. Jang, and S. Cho, "A 480-MHz to 1-GHz subpicosecond clock generator with a fast and accurate automatic frequency calibration in 0.13-μm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf.*, 2007, pp. 67–70.

[113] T.-H. Lin and Y.-J. Lai, "An agile VCO frequency calibration technique for a 10-GHz CMOS PLL," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 340–349, Feb. 2007.

[114] J. Shin and H. Shin, "A fast and high-precision VCO frequency calibration technique for wideband $\Delta\Sigma$ fractional-N frequency synthesizers," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 7, pp. 1573–1582, Jul 2010.

[115] W.-H. Chiu, Y.-H. Huang, and T.-H. Lin, "A dynamic phase error compensation technique for fast-locking phase-locked loops," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, pp. 1137–1149, Jun. 2010.

[116] W. Khalil, S. Shashidharan, T. Copani, S. Chakraborty, S. Kiaei, and B. Bakkaloglu, "A 700-μA 405-MHz all-digital fractional-N frequency-locked loop for ISM band applications," *IEEE Trans. Microw. Theory Tech.*, vol. 59, no. 5, pp. 1319–1326, May 2011.

[117] C.-C. Hung and S.-I. Liu, "A 40-GHz fast-locked all-digital phase-locked loop using a modified bang-bang algorithm," *IEEE Trans. Circuits Syst. II*, vol. 58, no. 6, pp. 321–325, Jun. 2011.

Aalto-DD 46/2014

ISBN 978-952-60-5637-1
ISBN 978-952-60-5638-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Micro- and Nanosciences
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS