ACSP · Analog Circuits And Signal Processing

Francesco Brandonisio
Michael Peter Kennedy

# Noise-Shaping All-Digital Phase-Locked Loops

## Modeling, Simulation, Analysis and Design

Springer

# Analog Circuits and Signal Processing



For further volumes:
http://www.springer.com/series/7381

Francesco Brandonisio
Michael Peter Kennedy

# Noise-Shaping All-Digital Phase-Locked Loops

## Modeling, Simulation, Analysis and Design

Springer

Francesco Brandonisio
Tyndall National Institute
Cork
Ireland

Michael Peter Kennedy
University College Cork
Cork
Ireland

ISSN 1872-082X ISSN 2197-1854 (electronic)
ISBN 978-3-319-03658-8 ISBN 978-3-319-03659-5 (eBook)
DOI 10.1007/978-3-319-03659-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013955900



Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To*
*Rocco, Nuccia*
*and*
*Rossana*

# Preface

All-Digital Phase Locked Loops (ADPLLs) have become very common in low cost and feature mobile phones. In recent years, extensive research activity on ADPLLs has focused on increasing the performance of ADPLLs, thus increasing the range of their possible applications. The theoretical noise performance of an ADPLL is limited by the quantization error of the TDC and the DCO. In the literature, noise shaping of the quantization error is one of the main techniques used to reduce the impact of the quantization error of the TDC and the DCO on the noise performance of an ADPLL. In this book, we present a framework to analyse, design, simulate and compare different ADPLL architectures with noise shaping TDCs and DCOs.

In Chap. 1, we summarize the main contributions of the book.

In Chap. 2, we review the operations of the main ADPLL architectures in terms of phase-to-digital conversion.

In Chap. 3, we review the main TDC architectures and relate their operations to quantizers and/or sigma-delta modulators.

In Chap. 4, we derive discrete-time models for the main ADPLL architectures and derive analytical equations for predicting the phase noise performance.

In Chap. 5, we show the advantages of noise shaping and dither by means of an analytical method in the time domain.

In Chap. 6, we focus on simulating ADPLLs as mixed-signal systems. We show that there is a tradeoff between accuracy and simulation time. We describe a simulation method in Simulink that can mitigate this tradeoff.

In Chap. 7, we discuss phase noise in more detail. We describe a procedure to model and extract the phase noise of a signal in Matlab.

This work was supported in part by Science Foundation Ireland under grant 08/IN.1/I854 and by Microelectronic Circuits Centre Ireland.

Cork, September 2013

Francesco Brandonisio
Michael Peter Kennedy

# Contents


**1 Introduction** . . . . . 1
References . . . . . 3

**2 Phase Digitization in All-Digital PLLs** . . . . . 7
2.1 Introduction. . . . . 7
2.2 Definitions of Integer and Fractional Phases . . . . . 7
2.3 Architectures of Phase-Difference Digitizing and Phase Digitizing ADPLLs. . . . . 10
2.4 Phase Digitization in a PFD-Plus-TDC-Based ADPLL . . . . . 12
2.4.1 Phase Difference with Integer Part Different from Zero . . . . . 15
2.4.2 Phase Difference with Integer Part Equal to Zero . . . . 20
2.5 Phase Digitization in TDC-Based ADPLL. . . . . 20
2.5.1 Phase Difference with Integer Part Different from Zero . . . . . 21
2.5.2 Phase Difference with Integer Part Equal to Zero . . . . 23
2.6 Phase Digitization in a Flip-Flop-Based ADPLL . . . . . 24
2.7 Phase Digitalization in an Accumulator-Based ADPLL. . . . . 26
2.7.1 Phase Difference with Integer Part Different from Zero . . . . . 27
2.7.2 Phase Difference with Integer Part Equal to Zero . . . . 31
2.8 Fractional-N Operations . . . . . 32
2.9 Comparison. . . . . 33
2.9.1 Phase-to-Digital Conversion . . . . . 34
2.9.2 Metastability . . . . . 34
2.9.3 TDC Dynamic Range. . . . . 36
2.10 Additional Design Considerations . . . . . 36
2.10.1 Introducing Digital-Sigma-Delta-Modulators in the Architectures . . . . . 36
2.10.2 Selecting the Clock of the Digital Filter . . . . . 37
2.11 Conclusions. . . . . 37
References . . . . . 38

**3 A Unifying Framework for TDC Architectures** . . . . . 39
3.1 Introduction . . . . . 39
3.2 Single Scale TDC and Flash Analog-to-Digital Converter . . . . . 39
3.3 Notation . . . . . 41
3.4 The Delay-Line-Based TDC . . . . . 41
3.5 The Oscillator-Based TDC (Plus Delay Line TDC) . . . . . 43
3.6 Dual Scale Time-to-Digital Converters and the Vernier Method . . . . . 45
3.6.1 The Vernier Method . . . . . 45
3.6.2 The Vernier Method as a Case of Sigma-Delta Modulation . . . . . 48
3.7 The Vernier Delay-Line-Based TDC . . . . . 49
3.8 The Vernier Oscillator-Based TDC . . . . . 51
3.9 The Pulse-Shrinking TDC . . . . . 54
3.10 The Gated-Ring-Oscillator-Based TDC . . . . . 56
3.11 TDC Behavioral Models . . . . . 61
3.12 Comparison . . . . . 64
3.13 Conclusion . . . . . 66
References . . . . . 66

**4 Analytical Predictions of Phase Noise in ADPLLs** . . . . . 69
4.1 Introduction . . . . . 69
4.2 Phase Noise . . . . . 69
4.3 Models of Time-to-Digital Converters . . . . . 72
4.4 DCO Model . . . . . 75
4.5 Digital Filter . . . . . 82
4.6 Full ADPLL Model . . . . . 84
4.7 Mixed-Signal Models . . . . . 88
4.8 An Example TDC-Based ADPLL . . . . . 90
4.9 Simulations with Machine-Precision TDC and DCO . . . . . 91
4.10 Simulations with a Finite-Resolution TDC . . . . . 92
4.11 Simulations with a Noise-Shaping TDC . . . . . 95
4.12 Simulations with a Finite-Resolution TDC and a Finite-Resolution DCO . . . . . 97
4.13 Conclusion . . . . . 98
References . . . . . 98

**5 Advantages of Noise Shaping and Dither** . . . . . 101
5.1 Introduction . . . . . 101
5.2 Quantizers and Sigma-Delta Modulators with Dither . . . . . 101
5.3 Comparison Between Zeroth and First-Order Shaped Dither . . . 107
5.3.1 Comparison Between a Quantizer and a First-Order Sigma-Delta Modulator . . . . . 109
5.4 Conclusion . . . . . 110
References . . . . . 110

**6 Efficient Modeling and Simulation of Accumulator-Based ADPLLs** . . . . . 111
6.1 Introduction . . . . . 111
6.2 Mixed Signal Systems . . . . . 111
6.3 Simulation Tools and Solvers of Differential Equations . . . . . 112
6.3.1 Continuous-Time and Discrete-Event Simulators . . . . . 112
6.3.2 Commercial Simulators . . . . . 113
6.4 Design of Mixed-Signal Systems on Chip . . . . . 114
6.5 Simulation Issues with All-Digital PLLs . . . . . 115
6.5.1 Generating Events at Fixed Instants . . . . . 115
6.5.2 Variation of the Time Step . . . . . 116
6.5.3 Quantization of Time . . . . . 116
6.5.4 Long Time Required to Complete a Simulation . . . . . 116
6.5.5 Multiple Clocks and Metastability . . . . . 117
6.6 Efficient Model of an Accumulator-Based ADPLL . . . . . 117
6.7 Efficient Model of a Reference Oscillator . . . . . 121
6.8 Simple Efficient DCO Model . . . . . 124
6.9 Improved Efficient DCO Model . . . . . 126
6.10 Modeling Strategy for the Other Building Blocks of an ADPLL . . . . . 131
6.11 The Retiming Block . . . . . 132
6.12 Digital Filter . . . . . 133
6.13 Accumulator and Phase Sampler . . . . . 134
6.14 TDC . . . . . 136
6.15 Simulation Results . . . . . 138
6.16 Increasing the Complexity of the DCO and the Phase Sampler . . . . . 139
6.17 Conclusion . . . . . 140
References . . . . . 140

**7 Modelling and Estimating Phase Noise with Matlab** . . . . . 143
7.1 Introduction . . . . . 143
7.1.1 Modelling Phase Noise . . . . . 143
7.1.2 Modelling Jitter . . . . . 146
7.1.3 Normalizations . . . . . 151
7.2 Conclusion . . . . . 152
References . . . . . 152

**Appendix A** . . . . . 153

**Index** . . . . . 177

# Acronyms

| | |
|---|---|
| ADC | Analog-to-Digital Converter |
| ADPLL | All-Digital Phase Locked Loop |
| DC | Direct Current; here it denotes a constant signal |
| DCO | Digitally Controlled Oscillator |
| DNL | Differential Nonlinearity |
| EMF | Embedded Matlab Function |
| FPGAy | Field Programmable Gate Array |
| GRO | Gated Ring Oscillator |
| INL | Integral Nonlinearity |
| LTI | Linear Time Invariant |
| PFD | Phase Frequency Detector |
| PLL | Phase Locked Loop |
| PSD | Power Spectral Density |
| TDC | Time-to-Digital Converter |
| VCO | Voltage-Controlled Oscillator |

# Chapter 1
# Introduction

An All Digital Phase Locked Loop (ADPLL) is an alternative to a traditional Phase Locked Loop (PLL) for implementation in nanoscale digital CMOS, especially as part of a system-on-chip (SoC) [1, 2]. One of the key advantages of ADPLLs over their analog counterparts is that they remove the need for large capacitors within the loop filter by utilizing digital circuits to achieve the desired filtering function. The resulting area savings are important for achieving a low-cost solution. Moreover, the phase error signal in an ADPLL is a digital word that is produced by means of digital circuits such as accumulators, samplers and Time-to-Digital Converters [3, 4]. As a consequence, a much more attractive, mostly digital, design flow is achieved [2]. Furthermore, the essentially digital architecture of an ADPLL can be augmented with reconfigurable gains and a filter [5, 6], or with a software-assisted digital processor for calibration [7].

In the literature of recent years, there has been a significant effort to improve the performance of ADPLLs in terms of the output frequency [6, 8], resolution [9], bandwidth [10–12], locking speed [13, 14], phase noise [15–17], and power consumption [18].

In this book, we focus on ADPLLs that include TDCs with noise-shaping of the quantization error. We explain how to design an ADPLL, analyse its noise performance and realize behavioral models that require the minimum possible simulation time. We also review the main TDC architectures in order to highlight the advantages of a first-order noise-shaping TDC. We show how to relate the operation of a TDC to a quantizer and/or a first-order sigma-delta modulator. Furthermore, we determine analytically the precisions of a quantizer and a sigma-delta modulator when followed by a moving average filter in terms of the maximum difference between the input and the output when this difference is bounded.

First-order noise-shaping of the quantization error is produced by a system that can be represented by a first-order sigma-delta modulator. During our analysis, we show that there is an *inherent* sigma-delta modulation in both Vernier TDCs and accumulator-based ADPLLs. In fact, we demonstrate that the models of a Vernier TDC and an accumulator-based ADPLL can be based on sigma-delta modulators. In

F. Brandonisio and M. P. Kennedy, *Noise-Shaping All-Digital Phase-Locked Loops*, Analog Circuits and Signal Processing, DOI: 10.1007/978-3-319-03659-5_1,

particular, we use the equations of a sigma-delta modulator to implement an efficient model of an accumulator-based ADPLL.

The book is organized as follows:

In Chap. 2, we review the operating principles of the main topologies of ADPLLs (PFD-plus-TDC-based, TDC-based and accumulator-based ADPLLs) in terms of the integer and fractional parts of the phase difference. We also mention the flip-flop-based ADPLL which can be considered as a special case of a TDC-based ADPLL. We show models that describe the phase-to-digital conversion in each ADPLL architecture. We also discuss possible strategies to clock the digital filter in the various ADPLL architectures. We show how to modify the ADPLL architectures to synthesize a fractional ratio between the frequencies of the reference oscillator and the DCO. Finally, we compare the ADPLL architectures in terms of phase-to-digital conversion, TDC dynamic range, and metastability. The analytical approach used in this chapter is general and it can be extended to any modified ADPLL architecture that can be obtained from those we have studied.

In Chap. 3, we briefly review the main TDC architectures that have been published in the literature. We introduce notation for comparing different TDC architectures in terms of their operating principles and time resolution. By comparing different TDC architectures, we show that a first-order noise-shaping TDC is an interesting architecture for digital systems because it allows a tradeoff between high time resolution and speed. We also show how the Vernier method can be considered as a special case of sigma-delta modulation. To our knowledge, the Vernier method is recognized as a particular case of sigma-delta modulation for the first time in this work. A Vernier-TDC model based on a sigma-delta modulator is also new in the literature, to our knowledge. There are already reviews of TDC architectures such as [19] and [20]. However, in this review, a *single* notation is used to derive models based on quantizers and sigma-delta modulators for the main TDC architectures from their respective timing diagrams. The same notation allows a simple comparison between different architectures in terms of time resolution.

In Chap. 4, we focus on the noise performance of the main architectures of ADPLLs. We derive analytical predictions of the phase noise in TDC-based and accumulator-basedADPLLs with noise-shaping TDCs and a DCO driven by a sigma-delta modulator. In order to derive analytical predictions for the ADPLL phase noise, we first explain how to calculate the phase noise of the DCO when its input is known. Then we derive linear models associated with the building blocks in an ADPLL. We use the linear models of the building blocks of an ADPLL to develop a linear model of the full ADPLL. We derive analytical predictions of the ADPLL phase noise from the linear ADPLL model. Finally, we compare Matlab simulations and analytical predictions for an example TDC-based ADPLL architecture. We also show that our results are in very good agreement with predictions obtained by means of the "PLL Design Assistant" [21] which is an automatic design tool for PLL. By contrast with the "PLL Design Assistant" program, which is compiled and therefore cannot be modified by the user, the Matlab scripts that we present are ready to be edited. Hence, the Matlab scripts in this chapter represent a complementary learning tool that gives direct insight into the design equations.

In Chap. 5, we determine the precision of the systems "Quantizer plus Moving Average Filter" and "Sigma-Delta Modulator plus Moving Average Filter" with dither. We show analytically that the difference between the input and output of a "Sigma-Delta Modulator plus Moving Average Filter" is smaller than that of an equivalent "Quantizer plus Moving Average Filter". The analytical results derived in this chapter are important in order to understand how to exploit noise shaping in ADPLLs. Notice that we determine the precision of systems comprising sigma-delta modulation and dither followed by moving average filters in terms of the maximum difference between the input and the output and the variance of the output in the *time domain*. In the literature, it is common to analyse the advantages of sigma-delta modulation and dither in the *frequency domain* [22]. However, the analysis in the frequency domain is usually based on the assumptions that the quantization error of a sigma-delta modulator is white and independent of the input. This white noise approximation is not valid when the output of a sigma-delta modulator exhibits tones. The approach that is presented in this chapter is an alternative to the standard frequency domain analysis and does not require the white noise approximation.

In Chap. 6, we discuss how to simulate an ADPLL with fully nonlinear behavioral models. We show how to realize an efficient behavioral model of an ADPLL that produces the *minimum* number of samples during a simulation. The equations that we use to implement the efficient model of an ADPLL are related to a sigma-delta modulator. We report example C- and Matlab code that can be used to implement an efficient Simulink model of an ADPLL. We also illustrate how to realize a Simulink S-function that controls the simulation loop while the simulation is running. The modelling approach for ADPLLs that is detailed in this chapter is similar to that described by Staszewski et al. [23]. The technique reported in [23] describes how to realize event-driven models in Verilog-AMS. Models that are event-driven can be built in Verilog and Verilog-AMS by means of the command "timer". In this chapter, we show how to realize event-driven models in Simulink. The Simulink models that we describe are simple and can be used as learning tools to understand how to deal with the problems that are associated with modeling ADPLLs.

In Chap. 7, we show how to model and calculate the phase noise of an oscillator in Matlab. The approach presented has been adopted in Chap. 4 to calculate the phase noise of an ADPLL. The material included in this chapter refers mostly to Kundert's work on model oscillators and signals with phase noise in Verilog-AMS [24]. However, we focus more on the steps that are necessary to model a noisy signal and to extract its phase noise. The steps of an example noise extraction procedure are clearly illustrated with Matlab scripts.

## References

1. M. Kumm, H. Klingbeil, P. Zipf, An FPGA-Based Linear All-Digital Phase-Locked Loop, IEEE Trans. Circuits Sys. I: Regular Papers, vol. 57, no. 9, pp. 2487–2497, Sep. 2010.
2. R. B. Staszewski, State-of-the-Art and Future Directions of High-Performance All-Digital Frequency Synthesis in Nanometer CMOS, IEEE Trans. Circuits Sys. I: Regular Papers, vol. 58, no. 7, pp. 1497–1510, Jul. 2011.

3. R. B. Staszewski, P. T. Balsara, All-Digital Frequency Synthesizer in Deep-Submicron CMOS, John Wiley & Sons Inc., 2006.
4. R. B. Staszewski, P, T. Balsara, Phase-Domain All-Digital Phase-Locked Loop, IEEE Trans. Circuits Sys. II: Express, Briefs, vol. 52, no. 3, Mar. 2005.
5. D.-S. Kim, H. Song, T. Kim, S. Kim, D.-K. Jeong, A 0.3–1.4 GHz All-Digital Fractional-N PLL With Adaptive Loop Gain Controller, IEEE J. Solid-State Circuits, vol. 45, no. 11, pp. 2300–2311, Nov. 2010.
6. S.-Y. Yang, W.-Z. Chen, 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.
7. R. Staszewski, R. B. Staszewski, T. Jung, T. Murphy, I. Bashir, O. Eliezer, K. Muhammad, M. Entezari Software Assisted Digital RF Processor (DRP$^{TM}$) for Single-Chip GSM Radio in 90 nm CMOS, IEEE J. Solid-State Circ., vol. 45, no. 2, pp. 276–288, Feb. 2010.
8. C.-C. Hung, S.-I. Liu, A 40-GHz Fast-Locked All-Digital Phase-Locked Loop Using a Modified Bang-Bang Algorithm, IEEE Trans. Circuits Sys. II: Express Briefs, vol. 58, no. 6, pp. 321–325, Jun. 2011.
9. S.-K. Lee, Y.-H. Seo, H.-J. Park, J.-Y. Sim, A 1 GHz ADPLL with a 1.25 ps Minimum-Resolution Sub-Exponent TDC in 0.18 $\mu$m CMOS, IEEE J. Solid-State Circuits, vol. 45, no. 12, pp. 2874–2881, Dec. 2010.
10. M. Zanuso, S. Levantino, C. Samori, A. L. Lacaita, A Wideband 3.6 GHz Digital $\Delta\Sigma$ Fractional-N PLL With Phase Interpolation Divider and Digital Spur Cancellation, IEEE J. Solid-State Circuits, vol. 46, no. 3, pp. 627–638, Mar. 2011.
11. E. Temporiti, C. Weltin-Wu, D. Baldi, M. Cusmai, F. Svelto, A 3.5 GHz Wideband ADPLL With Fractional Spur Suppression Through TDC Dithering and Feedforward Compensation, IEEE J. Solid-State Circuits, vol. 45, no. 12, pp. 2723–2736, Dec. 2010.
12. E. Temporiti, C. Weltin-Wu, D. Baldi, R. Tonietto, 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.
13. X. Chen, J. Yang, L.-X. Shi, A Fast Locking All-Digital Phase-Locked Loop via Feed-Forward Compensation Technique IEEE Trans. Very Large Scale Int. (VLSI) Sys., vol. 19, no. 5, pp. 857–868, May. 2011.
14. C.-T. Wu, W.-C. Shen, W. Wang, A.-Y. Wu, A Two-Cycle Lock-In Time ADPLL Design Based on a Frequency Estimation Algorithm, IEEE Trans. Circuits Sys. II: Express, Briefs, vol. 57, no. 6, Jun. 2010.
15. H.-J. Hsu, S.-Y. Huang, A Low-Jitter ADPLL via a Suppressive Digital Filter and an Interpolation-Based Locking Scheme IEEE Trans. Very Large Scale Int. (VLSI) Sys., vol. 19, no. 1, pp. 165–170, Jan. 2011.
16. T. Tokairin, M. Okada, M. Kitsunezuka, T. Maeda, M. Fukaishi, A 2.1–2.8-GHz Low-Phase-Noise All-Digital Frequency Synthesizer With a Time-Windowed Time-to-Digital Converter, IEEE J. Solid-State Circuits, vol. 45, no. 12, pp. 2582–2590, Dec. 2010.
17. C.-M. Hsu, M. Z. Straayer, 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.
18. L. Xu, S. Lindfors, K. Stadius, Jussi RyynRnen, 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.
19. G.W. Roberts, M. Ali-Bakhshian, A Brief Introduction to Time-to-Digital and Digital-to-Time Converters, IEEE Trans. Circuits Sys. II: Express, Briefs, vol. 57, no. 3, Mar. 2010.
20. J. Kalisz, Review of Methods for Time Interval Measurements with Picosecond Resolution, Metrologia, vol. 41, no. 1, pp. 1732, Feb. 2004.
21. PLL Design Assistant Program, a design tool developed by Mike Perrott and available at http://www.cppsim.com Accessed in Nov. 2012

22. R. Schreier, G. C. Temes, Understanding Delta-Sigma Data Converters, IEEE Press, John Wiley & Sons Inc., 2005.
23. R.B. Staszewski, P.T. Balsara, C. Fernando, Event-driven Simulation and modeling of phase noise of an RF oscillator, IEEE Trans. Circuits Sys. I: Regular Papers, vol. 52, no. 4, Apr. 2005.
24. K. Kundert, Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers, Designer's Guide Consulting Inc., downloadable at http://www.designers-guide.org/analysis/PLLnoise+jitter.pdf. Last access in October 2012

# Chapter 2
# Phase Digitization in All-Digital PLLs

## 2.1 Introduction

In this chapter, we review the operating principles of the main topologies of ADPLLs (PFD-plus-TDC-based, TDC-based and accumulator-based ADPLLs) in terms of the integer and fractional parts of the phase difference. We also mention the flip-flop-based ADPLL which can be considered as a particular case of a TDC-based ADPLL. We show models that describe the phase-to-digital conversion in each ADPLL architecture when the integer part of the phase difference is equal to or different from zero. We show that a flip-flop based ADPLL can be viewed as the ADPLL architecture with the simplest phase-to-digital conversion. We also discuss possible strategies to clock the digital filter in the various ADPLL architectures. We show how to modify the ADPLL architectures to synthesize a fractional ratio between the frequencies of the reference oscillator and the DCO. Finally, we compare the ADPLL architectures in terms of phase-to-digital conversion, TDC dynamic range, and metastability.

## 2.2 Definitions of Integer and Fractional Phases

The phase $\boldsymbol{\Phi}(t)$ of a signal at the instant $t$ is the integral of the instantaneous frequency over the time interval $[0, t]$, as shown by the following equation:

$$\boldsymbol{\Phi}(t) = \int_{\tau=0}^{t} f(\tau) d\tau + \boldsymbol{\Phi}_0, \tag{2.1}$$

where $f(t)$ and $\boldsymbol{\Phi}_0$ are the instantaneous frequency and the initial phase of the signal. According to Eq. (2.1), the phase of a signal is determined when the initial phase and the frequency $f(t)$ over the interval $[0, t]$ are known. In order to simplify our analysis, we assume that the signal with unknown frequency can be associated with

F. Brandonisio and M. P. Kennedy, *Noise-Shaping All-Digital Phase-Locked Loops*, Analog Circuits and Signal Processing, DOI: 10.1007/978-3-319-03659-5_2,

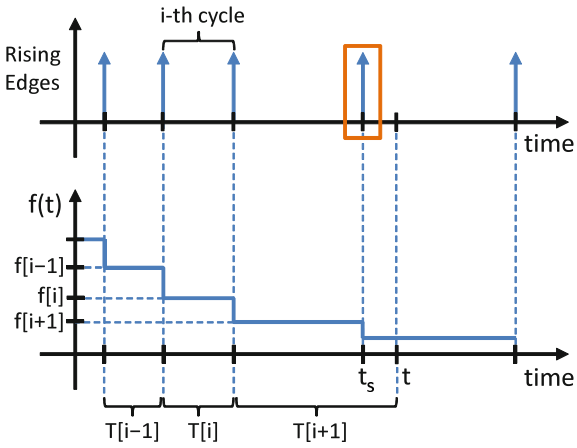


**Fig. 2.1** Timing diagram showing the cycles and the instantaneous frequency of a signal

cycles. The cycles are determined by choosing a reference level which is crossed by the signal. The beginning and the end of a cycle are determined by two consecutive crossings. Without loss of generality, we assume that the rising edges of a signal are associated with the beginnings and the ends of the cycles. Moreover, we assume that the frequency of the signal is *constant* and equal to $f[i]$ over the $i$-th cycle. Hence, the duration of the $i$-th cycle is equal to the period $T[i] = 1/f[i]$ of the signal. The frequency $f[i]$ can be obtained by measuring the duration $T[i]$ of the $i$-th cycle. Without loss of generality, we assume that the rising edges of a signal determine the durations of the cycles that have to be measured. Figure 2.1 shows the cycles and the values of the instantaneous frequency of an example signal.

When the frequency $f(t)$ is a piecewise constant function of the type shown in Fig. 2.1, Eq. (2.1) can be simplified, as we will show in the following. We first write Eq. (2.1) as:

$$\boldsymbol{\Phi}(t) = \int_{\tau=0}^{t_s} f(\tau)d\tau + \int_{\tau=t_s}^{t} f(\tau)d\tau, \tag{2.2}$$

where $t_s$ is the instant associated with the previous rising edge with respect to $t$, as shown in Fig. 2.1. There are two integrals in Eq. (2.2). By considering that $f(t)$ is equal to $1/T[k]$ between two consecutive rising edges of the $k$-th cycle, the first integral in Eq. (2.2) gives:

$$\int_{\tau=0}^{t_s} f(\tau)d\tau = \sum_{i=1}^{n} f[i] \cdot T[i] = n, \tag{2.3}$$

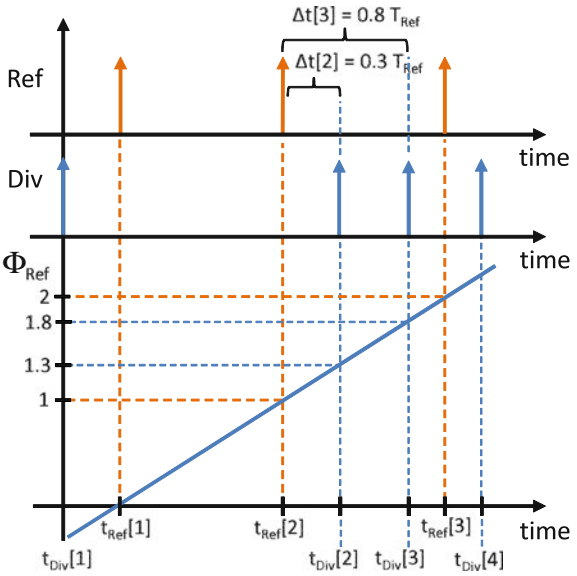


**Fig. 2.2** Phase of the signal *Ref* expressed in cycles

where $n$ is the number of *complete* cycles of the signal over the time interval $[0, t]$. We define the integer part of the phase $\boldsymbol{\Phi}_{int}(t)$ of a signal as the number of *complete* cycles of the signal over the time interval $[0, t_s]$.

By considering that $f(t)$ is equal to $1/T[n+1]$ over the time interval $[t_s, t]$, the second integral in Eq. (2.2) gives:

$$\int_{\tau=t_s}^{t} f(\tau)d\tau = \frac{(t-t_s)}{T[n+1]}. \tag{2.4}$$

We define the fractional part of the phase $\boldsymbol{\Phi}_{frac}(t)$ as the ratio $(t-t_s)/T[n+1]$. We conclude that the phase $\boldsymbol{\Phi}(t)$ of a signal can be written as:

$$\boldsymbol{\Phi}(t) = \boldsymbol{\Phi}_{int}(t) + \boldsymbol{\Phi}_{frac}(t). \tag{2.5}$$

Assume two signals, *Ref* and *Div* whose cycles are defined by their respective rising edges. Moreover, we assume that the periods of *Ref* and *Div* are constant and variable with respect to time, respectively. This assumption is consistent with the operations of an ADPLL that we will analyse in the next sections.

By means of Eqs. (2.3) and (2.4), we can define the integer and fractional parts of the phase $\boldsymbol{\Phi}_{Ref}$ of *Ref* at the instants $t_{Div}[n]$ at which the edges of *Div* occur, as shown in Fig. 2.2. The integer part of $\boldsymbol{\Phi}_{Ref}$ at the instant $t_{Div}[n]$ is determined by the number of previous edges of *Ref*, where $n$ is the index associated with the edges of *Div*. The fractional part of $\boldsymbol{\Phi}_{Ref}$ at $t_{Div}[n]$ is equal to $\Delta t[n]/T_{Ref}$, where $\Delta t[n]$ is the

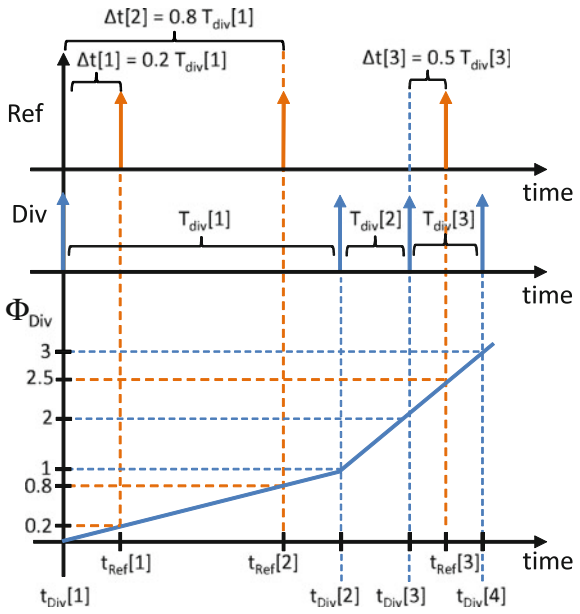


**Fig. 2.3** Phase of the signal *Div* expressed in cycles

duration of the time interval between $t_{Div}[n]$ and the instant of the previous edge of *Ref*, as shown in Fig. 2.2.

It is important to notice from Eqs. (2.3) and (2.4) that we can also define the phase $\boldsymbol{\Phi}_{Div}$ of *Div* at the instants $t_{Ref}[n]$ at which the edges of *Ref* occur, as shown in Fig. 2.3.

The integer part of $\boldsymbol{\Phi}_{Div}$ at the instant $t_{Ref}[n]$ is equal to the number of the previous edges of *Ref*, where $n$ is the index associated with the edges of *Ref*. The fractional part of $\boldsymbol{\Phi}_{Div}$ at $t_{Ref}[n]$ is equal to $\Delta t[n]/T_{Div}[k]$, where $\Delta t[n]$ is the duration of the time interval between $t_{Ref}[n]$ and the instant of the previous edge of *Div*, and $T_{Div}[k]$ is the duration of the $k$-th cycle of *Div* which includes $t_{Ref}[n]$, as shown in Fig. 2.3.

Finally, the phase difference between *Ref* and *Div* can be determined from Figs. 2.2 and 2.3, as shown in Fig. 2.4.

Considerations of the fractional and integer phases of *Ref* and *Div* and the definition of the phase difference between two signals are fundamental to understanding and comparing the operations of the ADPLL architectures that we will consider in the next sections.

## 2.3 Architectures of Phase-Difference Digitizing and Phase Digitizing ADPLLs

The operation of an ADPLL is based on the association of a digital word with the phase difference between two signals. In order to realize this association, it is possible to follow two approaches. The first approach is to produce an analog measurement