

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.

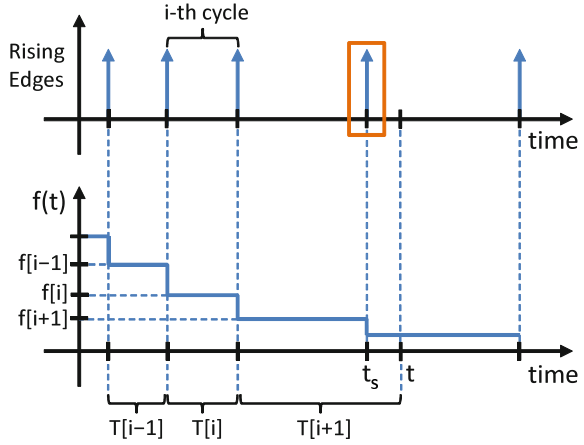
2.2 Definitions of Integer and Fractional Phases

The phase $\Phi(t)$ 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:

$$\Phi(t) = \int_{\tau=0}^t f(\tau) d\tau + \Phi_0, \quad (2.1)$$

where $f(t)$ and Φ_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

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.

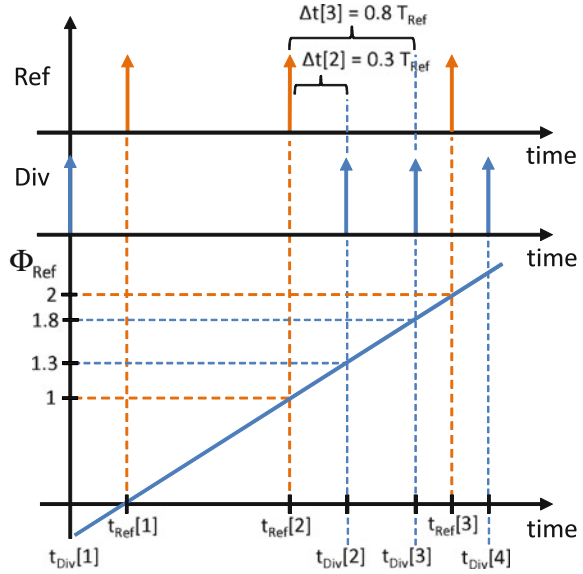
When the frequency $f(t)$ is a piecewise constant function of the type shown in Fig. 2.1, Eq. (2.1) can be simplified, as we will show in the following. We first write Eq. (2.1) as:

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

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:

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

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 $\Phi_{int}(t)$ 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:

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

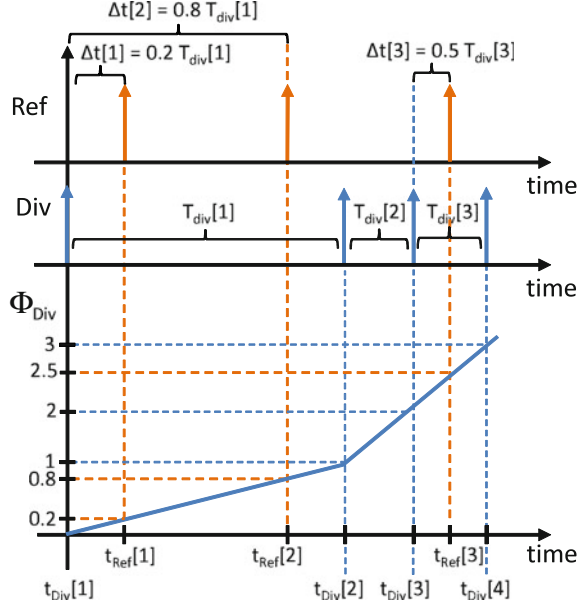
We define the fractional part of the phase $\Phi_{frac}(t)$ as the ratio $(t - t_s)/T[n+1]$. We conclude that the phase $\Phi(t)$ of a signal can be written as:

$$\Phi(t) = \Phi_{int}(t) + \Phi_{frac}(t). \quad (2.5)$$

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 Φ_{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 Φ_{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 Φ_{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 Φ_{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 Φ_{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 Φ_{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.

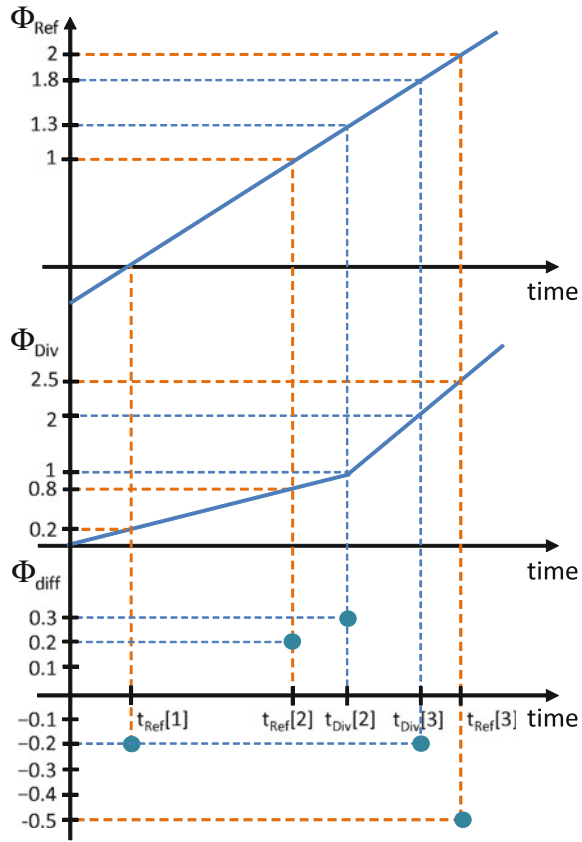
Finally, the phase difference between *Ref* and *Div* can be determined from Figs. 2.2 and 2.3, as shown in Fig. 2.4.

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

Fig. 2.4 Sampled phase difference between Ref and Div at time instants $t_{Ref}[i]$ and $t_{Div}[i]$



of the phase difference and then to convert the analog measurement to a digital word [1], as illustrated in Fig. 2.5a. ADPLLs that use this approach can be classified as “phase-difference digitizing” ADPLLs.

The second approach to realize an ADPLL is to produce digital words related to the phases of the signals and then to calculate the difference between the digitized phases [1], as shown in Fig. 2.5b. ADPLLs that use this approach can be classified as “phase digitizing” ADPLLs. The block diagram in Fig. 2.5a can be related to the operations of PFD-plus-TDC-based, TDC-based and flip-flop-based ADPLLs. The block diagram in Fig. 2.5b is associated with accumulator-based ADPLLs.

In the next sections, we will show how the integer and fractional parts of the phases of the signals in PFD-plus-TDC-based, TDC-based, flip-flop-based and accumulator-based ADPLL are converted to digital words.

Fig. 2.5 ADPLL block diagrams in which **a** the phase difference is first associated with an analog signal and then converted to a digital word and **b** the phases of the reference oscillator and the DCO are converted to digital words and then subtracted

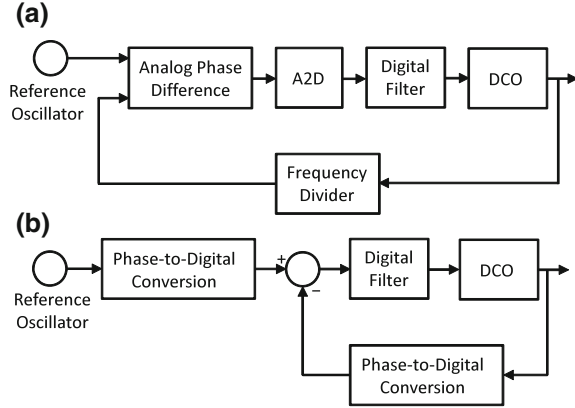
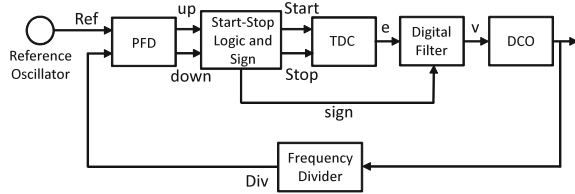


Fig. 2.6 Block diagram of a PFD-plus-TDC-based ADPLL



2.4 Phase Digitization in a PFD-Plus-TDC-Based ADPLL

The architecture of a PFD-plus-TDC-based ADPLL is shown in Fig. 2.6 [2–4]. Some works in the literature [2–4] do not show the block labelled “Start–Stop Logic and Sign” in Fig. 2.6. In these works, the operations of the block “Start–Stop Logic and Sign” are included in the TDC. We separate the block “Start–Stop Logic and Sign” from the TDC in order to simplify the description of the operation of a PFD-plus-TDC-based ADPLL.

In a PFD-plus-TDC-based ADPLL, the phase difference between the signals *Ref* and *Div* is measured by means of a Phase-Frequency-Detector (PFD).

Figure 2.7a, b shows the state diagram and the implementation with logic gates of a tristate PFD, respectively. The blocks labelled “DFF” in Fig. 2.7b represent D-flipflops.

In the PFD-plus-TDC-based ADPLL in Fig. 2.6, the block labelled “Start–Stop Logic” produces the *Start* and *Stop* signals for a TDC from the up and down pulses of a PFD.

Example signals *up* and *down* are shown in Fig. 2.8. We assume that the block “Start–Stop Logic” starts and stops the TDC when the PFD starts and stops producing a pulse, respectively. The signals “Start” and “Stop” are equal to “*up* OR *down*” and “(NOT *up*) AND (NOT *down*)”, respectively, as shown in Fig. 2.8. The signal *sign*

Fig. 2.7 **a** State diagram and **b** implementation with logic gates of a tristate PFD

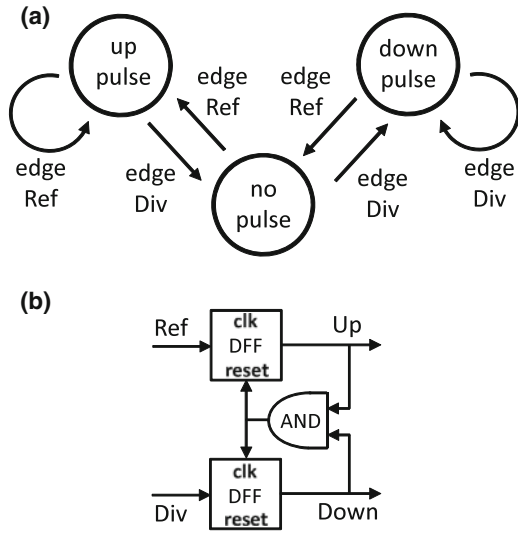
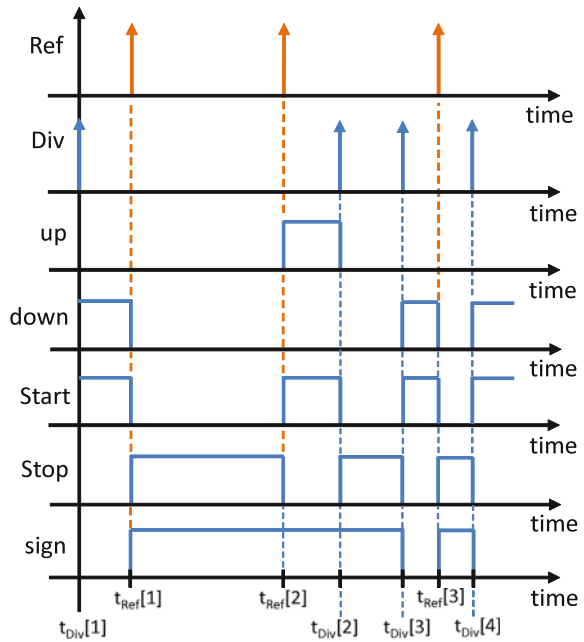
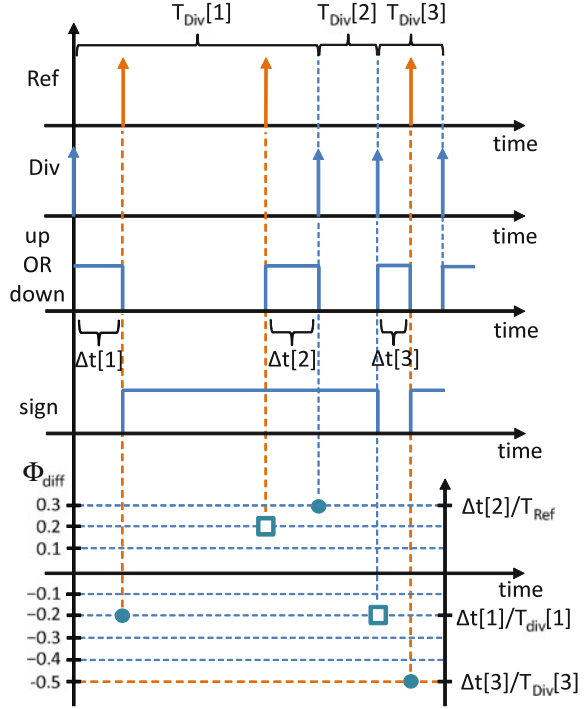


Fig. 2.8 Example timing diagrams showing the signals *up*, *down*, *Start*, *Stop* and *sign* in a PFD-plus-TDC-based ADPLL



is equal to “*up* OR ((NOT *up*) AND (NOT *down*))”; it is high during an up pulse and when the signals *up* and *down* are low at the same time. The signal *sign* is low during a down pulse.

Fig. 2.9 Phase difference, state of the PFD and TDC output. The *circles* represents the measurement produced by the TDC combined with the signal *sign*



The durations of the up and down pulses that are produced by the PFD in Fig. 2.6 provide an *analog* measure of the phase difference between the signals *Ref* and *Div*, as shown in Fig. 2.9.

The signal *sign* is high when the phase difference is zero or positive, as shown in Fig. 2.9. The solid circles in Fig. 2.9 represent the measurements produced by the TDC combined with the signal *sign*.

The open squares in Fig. 2.9 are associated with values of the phase difference related to the time intervals when the *up* and *down* signals are both low. In order to keep our analysis simple, we assumed that the TDC does not measure the duration of these time intervals. In Fig. 2.9, we consider the same signals *Ref*, *Div* and Φ_{diff} shown in Figs. 2.2, 2.3 and 2.4, respectively. The periods T_{Ref} and $T_{Div}[k]$ are constant and variable with respect to time, respectively. The index k in $T_{Div}[k]$ indicates the k -th cycle of *Div*. Moreover, we assume that the output of the TDC before normalization is a measure of $\Delta t[n]$ expressed in seconds, where n is the index associated with each pulse produced by the PFD. In order to obtain the phase difference between *Ref* and *Div*, the output of the TDC has to be normalized by T_{Ref} and $T_{Div}[k]$ when the *last edge* of the up or down pulse associated with $\Delta t[n]$ belongs to *Div* or *Ref*, respectively, according to the definitions given in Sect. 2.2 and as shown in Fig. 2.9. Hence, the output of the TDC in the architecture in Fig. 2.6 has to be normalized by a factor that alternates appropriately in time between T_{Ref} and $T_{Div}[k]$ in order to keep

the phase measurement consistent. This important design consideration is *missing in the literature* on PFD-plus-TDC-based ADPLLs.

The sign of the phase difference is positive and negative, when the signal *sign* is high and low, respectively, as shown in Fig. 2.9. We assume that the signal *sign* is high when the signals *up* and *down* are low at the same time. However, the TDC produces measurements of the modulus of the phase difference only after a falling edge of an up or down pulse. Hence, the level of the signal *sign* is not related to the measurement of the phase difference when the signals *up* and *down* are low at the same time, and it can be set to low or high arbitrarily. Finally, notice that the durations of the up and down pulses in a PFD-plus-TDC-based ADPLL are related to the modulus of the phase difference including *both integer and fractional* parts.

In the example timing diagrams in Fig. 2.9, the phase difference between *Ref* and *Div* has an integer part equal to zero. The fact that the integer part of the phase difference is equal to or different from zero changes drastically how a PFD-plus-TDC based ADPLL operates. In the following, we will first explain the operations of an PFD-plus-TDC based ADPLL when the integer part of the phase difference is different from zero. Then, we will resume the analysis of a PFD-plus-TDC based ADPLL in which the integer part of the phase difference is *equal* to zero.

2.4.1 Phase Difference with Integer Part Different from Zero

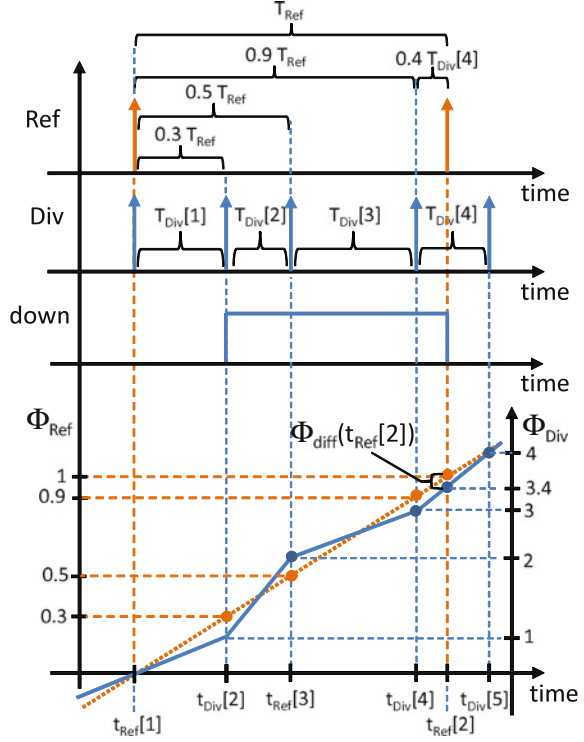
The phase difference between *Ref* and *Div* has a non-zero integer part when one or more complete cycles of *Div* or *Ref* are included between the rising and falling edges of an up or down pulse, respectively. In this section, we explain the operations of a PFD-plus-TDC-based ADPLL that can measure the integer part of the phase difference. However, the ADPLLs that are described in [2, 4] assume that the phase difference always has an integer part that is equal to zero. In fact, the implementation of an ADPLL becomes simpler when the ADPLL can measure only small values of phase difference. Hence, we illustrate in the following the complications that arise in a PFD-plus-TDC-based ADPLL that measures the integer and fractional parts of the phase difference.

Assume that T_{sum} is the sum of the durations of all these cycles.¹ The fractional part of the phase difference Φ_{diff} can be calculated by subtracting T_{sum} from the output of the TDC and then by normalizing the result, as explained in the initial part of this section. The integer part of the phase difference can be obtained by counting the number of complete cycles included in each up and down pulse. Let us illustrate this with an example.

Figure 2.10 shows example timing diagrams with the edges of the *Ref* and *Div*, and the respective phases Φ_{Ref} and Φ_{Div} . The sampled phases of Φ_{Ref} and Φ_{Div} at the instants $t_{Ref}[i]$ and $t_{Div}[i]$ are associated with the dashed and solid curves in Fig. 2.10, respectively. Notice that Φ_{Ref} and Φ_{Div} are associated with two different scales. The

¹ The durations of the periods of T_{Div} can be estimated from the input of the DCO.

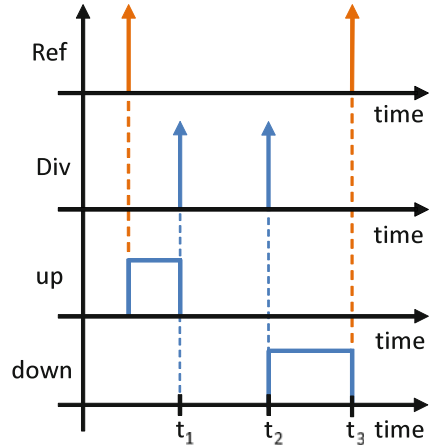
Fig. 2.10 Example timing diagram of the edges of *Ref* and *Div*



phase difference Φ_{diff} between Φ_{Ref} and Φ_{Div} can be obtained by subtracting Φ_{Div} from Φ_{Ref} . The phase difference $\Phi_{diff}(t_{Ref}[2])$ is equal to -2.4 and, hence, it has a *non-zero* integer part. In Fig. 2.10, the time interval Δt that the TDC measures is equal to $T_{Div}[2] + T_{Div}[3] + 0.4T_{Div}[4]$. The fractional part of $\Phi_{diff}(t_{Ref}[2])$ is given by $(\Delta t - T_{Div}[2] - T_{Div}[3]) / T_{Div}[4] = 0.4$. In a PFD-plus-TDC-based ADPLL the fractional part of Φ_{diff} can be obtained by subtracting T_{sum} from the output of the TDC before normalizing. By contrast, the integer part of $\Phi_{diff}(t_{Ref}[2])$ is given by the number of complete cycles of *Div* that are included between the rising and falling edges of the *down* pulse. In a PFD-plus-TDC based ADPLL, the integer part of Φ_{diff} can be produced by a counter that counts the number of complete cycles included in the up and down pulses. The signal e in Fig. 2.6 can be obtained as the sum of the fractional and integer parts of the phase difference.

The digital filter in a PFD-plus-TDC-based ADPLL can be clocked on the falling edges of the up and down pulses. However, the duration of the time interval between consecutive edges of the down and up pulses is variable. Hence, the period of the clock of the digital filter is variable when the digital filter is clocked by the falling edges of the up and down pulses. This may compromise the operation of the digital filter, as we will discuss in Sect. 2.10.2. In order to keep the period of the clock of the digital filter constant over time, it is possible to clock the digital filter with

Fig. 2.11 Example pair of up and down pulses included between two consecutive edges of *Ref*



the output of the reference oscillator. However, when the digital filter is clocked by the reference oscillator, only the last measurement of the phase difference produced by the TDC during a cycle of the reference oscillator is processed by the digital filter. For example, Fig. 2.11 shows a pair up and down pulses included between two consecutive edges of *Ref*. When the configuration of pulses in Fig. 2.11 happens in a PFD-plus-TDC-based ADPLL, the TDC produces measurements of the phase difference at the instants t_1 and t_3 in Fig. 2.11.

When the digital filter is clocked by the falling edges of the up and down pulses, both the phase difference measurements at the instants t_1 and t_3 are sampled and processed by the digital filter. However, only the phase difference measurement at the instant t_3 is sampled and processed by the digital filter when the digital filter is clocked by the reference oscillator.

It is convenient to illustrate the phase-to-digital conversion in a PFD-plus-TDC-based ADPLL by means of a behavioral block diagram in order to be able to show the effects of the TDC quantization on the ADPLL's noise performance and to compare a PFD-plus-TDC-based ADPLL with the other ADPLL architectures. We will compare the main ADPLL architectures in terms of quantization error in Sect. 2.9.

In fact, the system that comprises the PFD, the block “Start–Stop Logic and Sign” and the TDC in Fig. 2.6 is a phase-to-digital converter that can be represented by the model in Fig. 2.12. The block labelled “Ideal Phase Difference Measurement” in Fig. 2.12 produces an ideal measurement of the phase difference between *Ref* and *Div* at the end of each up and down pulse. The block “Ideal Phase Difference Measurement” represents the operations of an ideal PFD. The phase difference is expressed in cycles of *Ref* or *Div* when the last edge of the up or down pulse belongs to *Div* or *Ref*, respectively. The block labelled “Modulus and Conversion to Seconds” in Fig. 2.12 multiplies the modulus of the output of the block “Ideal Phase Difference Measurement” by the period of *Ref* or *Div* when the last edge of the up or down pulse belongs to *Div* or *Ref*, respectively. Notice that two up or down pulses with

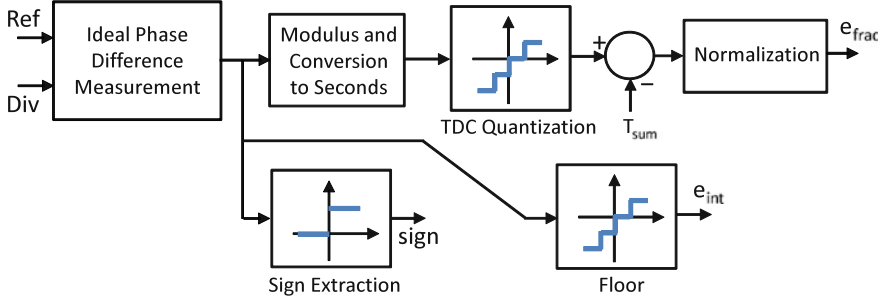


Fig. 2.12 Model of the system which comprises the PFD, the block “Start–Stop Logic and Sign” and the TDC in Fig. 2.6

different durations can produce the *same* phase difference at the output of the block “Ideal Phase Difference Measurement” when the last edges of the two pulses belong to *Ref* and *Div*, respectively. This is consistent with the fact that the same value of phase difference gives two different values at the output of the block “Conversion to Seconds” when the last edge of the up or down pulse belongs to *Div* or *Ref*.

The block labelled “TDC Quantization” in Fig. 2.12 represents the quantization effects of a TDC with finite time resolution res . The output *out* of the block “TDC Quantization” is given by:

$$out = res \cdot \text{floor}(in/res) \quad (2.6)$$

where $\text{floor}(\cdot)$ rounds to the largest integer smaller than the argument and res is the time resolution of the TDC. The block labelled “Normalization” in Fig. 2.12 divides its input by the period of *Ref* or *Div* when the last edge of the up or down pulse belongs to *Div* or *Ref*, respectively. Finally, the output of the block labelled “Sign Extraction” in Fig. 2.12 is equal to one when its input is positive or zero; otherwise it is zero.

The model in Fig. 2.12 can be modified to show the effect of a wrong normalization of the TDC output on the phase measurement in a PFD-plus-TDC-based ADPLL. We illustrate this effect by means of an example. Assume an example PFD-plus-TDC-based ADPLL in which the output of the TDC is normalized by T_{Ref} independently of the last edge of the up and down pulses. Moreover, we assume that the quantization in the TDC is negligible in order to keep our analysis simple. The block diagram in Fig. 2.12 simplifies to that in Fig. 2.13 when the output of the TDC is always normalized to the period T_{Ref} of *Ref* and the quantization noise in the TDC is negligible.

In the block diagram in Fig. 2.13, the signal e , which is the TDC output after normalization, is equal to the phase difference when the phase difference is positive. By contrast, the signal e in Fig. 2.13 is equal to $T_{Ref}/T_{Div}[k]$ times the phase difference when the phase difference is negative. The characteristic of the phase-to-digital conversion can be defined by the normalized TDC output as a function of the input

Fig. 2.13 Block diagram of the phase-to-digital conversion in a PFD-plus-TDC-based ADPLL with the output of the TDC normalized always by T_{Ref}

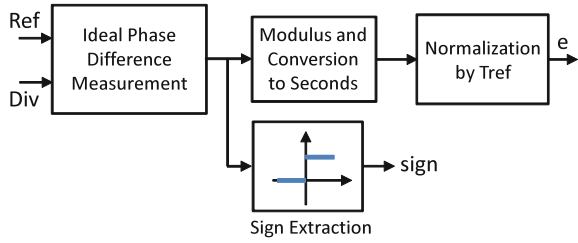
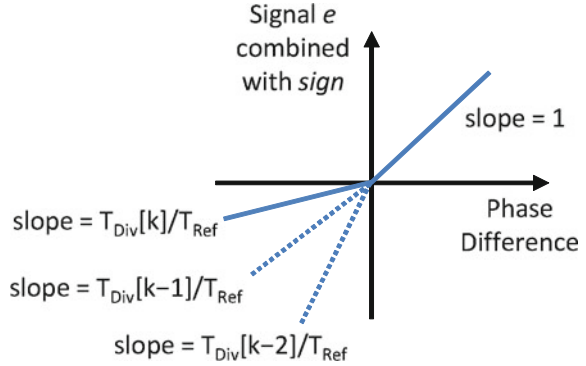


Fig. 2.14 Example characteristic of the phase-to-digital conversion associated with the block diagram in Fig. 2.13



phase difference. The TDC characteristic of the phase-to-digital conversion of the block diagram in Fig. 2.13 has a time-varying slope for negative values of the phase difference, as shown in Fig. 2.14.

The characteristic in Fig. 2.14 is nonlinear and time-varying. When the slope of the phase-to-digital characteristic is time-varying, the gain of the phase-to-digital characteristic changes over time. A time-varying gain in an ADPLL has to be taken into account in the design of the ADPLL in order to guarantee that the stability requirements are met for all possible values of the gain. Moreover, an ADPLL with a characteristic of the type in Fig. 2.14 shows different transient responses to positive and negative steps in the phase difference.

In order to avoid the nonlinear time-varying phase-to-digital characteristic in Fig. 2.14, it is necessary to normalize the output of the TDC by the period of *Ref* or *Div* when the falling edge of an up or down pulse is associated with an edge of *Div* or *Ref*, respectively. Hence, the periods of *Ref* and *Div* have to be known in order to avoid this nonlinear time-varying characteristic. Moreover, the calculation of T_{sum} in Fig. 2.12 requires that all complete periods of *Div* included in an up or down pulse are known. Unfortunately, the period of *Div* is not always known, so this nonlinear time-varying effect cannot be eliminated by normalizing the TDC output.

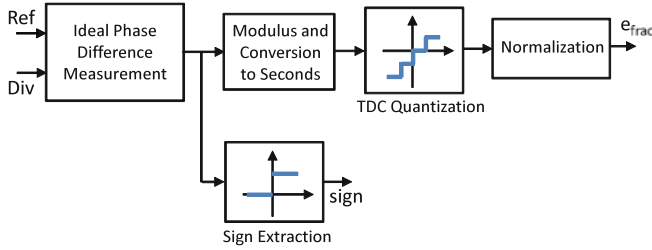


Fig. 2.15 Model of the system which comprises the PFD, the block “Start–Stop Logic and Sign” and the TDC in Fig. 2.6 when the phase difference has integer part equal to zero

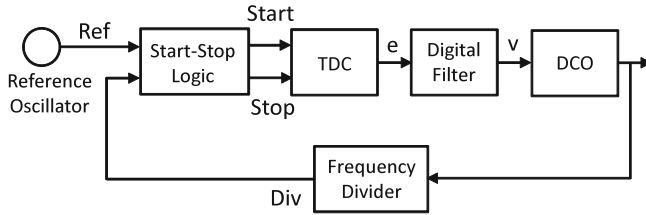


Fig. 2.16 Block diagram of a TDC-based ADPLL

2.4.2 Phase Difference with Integer Part Equal to Zero

The integer part of the phase difference is zero when there are one or two edges of *Div* between two consecutive edges of *Ref*. When there are one or two edges of *Div* between two consecutive edges of *Ref*, the up and down pulses produced by the PFD never include a complete period of *Ref* or *Div*. Hence, T_{sum} in the model in Fig. 2.12 is always equal to zero. The normalized output of the TDC in a PFD-plus-TDC-based ADPLL is equal to the phase difference when the integer part of the phase difference is zero. The model in Fig. 2.12 simplifies to that in Fig. 2.15. Hence, it is not necessary to subtract T_{sum} from the output of the TDC in a PFD-plus-TDC-based ADPLL in which the integer part of the phase difference is zero.

2.5 Phase Digitization in TDC-Based ADPLL

Figure 2.16 shows the block diagram of a TDC-based ADPLL [5]. The TDC in Fig. 2.16 starts and stops a measurement on a rising edge of *Ref* and on the next edge of *Div* [5], respectively. Hence, the output of the TDC normalized by T_{Ref} is equal to the fractional part of the phase of the reference oscillator on the edges of *Div*. The fractional part of the phase of the reference oscillator is equal to the fractional part of the phase difference Φ_{Ref} measured on the edges of *Div* when the integer parts

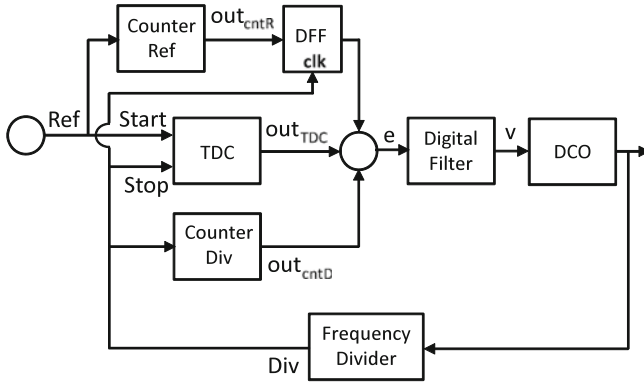


Fig. 2.17 Block diagram of a TDC-based ADPLL with counters to measure integer and fractional parts of the phase difference

of the phases of *Ref* and *Div* are equal on the edges of *Div*. Hence, the TDC-based ADPLL in Fig. 2.16 does not measure the integer part of the phase difference.

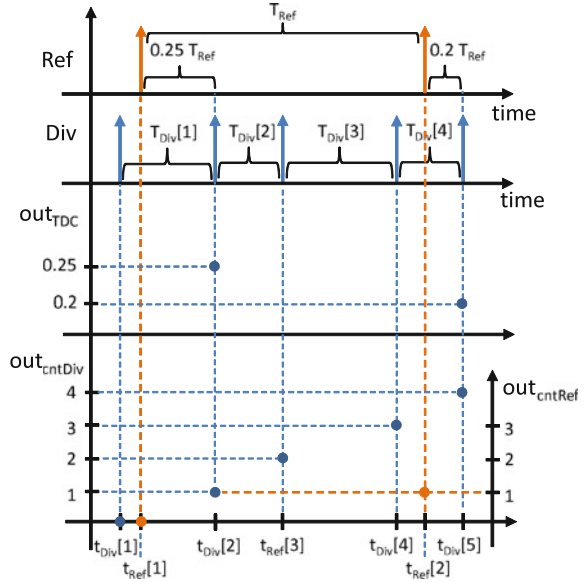
The digital filter in a TDC-based ADPLL can be clocked by the *Stop* signal in order to let the digital filter process the phase difference measurements as soon as the TDC produces them. The duration of the time interval between two consecutive edges of the *Stop* signal is varying over time. This may also lead to problems in the operation of the digital filter, as we will discuss in Sect. 2.10.2. By contrast, the period of the clock of the digital filter is constant when the digital filter is clocked by the reference oscillator.

Before analysing the architecture in Fig. 2.16, it is important to explain the operations of a TDC-based ADPLL that can measure the *integer and fractional* parts of the phase difference. The TDC-based ADPLL that we will first describe can measure the phase difference when the integer part of the phase difference is different from zero. Then, we will resume our analysis of the TDC-based ADPLL in Fig. 2.16 that can measure the phase difference only when the integer part of the phase difference is equal to zero.

2.5.1 Phase Difference with Integer Part Different from Zero

Two counters can be introduced in the ADPLL in Fig. 2.16 in order to measure the integer part of the phase difference, as shown in Fig. 2.17. The ADPLL in Fig. 2.17 measures the phase difference between *Ref* and *Div* at the instant associated with the first edge of *Div* that occurs after an edge of *Ref*. In order to explain how the phase difference is digitized in the ADPLL in Fig. 2.17, we consider the example timing diagrams of the edges of *Ref* and *Div* and their phases, as shown in Fig. 2.18.

Fig. 2.18 Example timing diagrams of the edges of *Ref* and *Div* and the outputs of the TDC and the counters in the ADPLL in Fig. 2.17



Notice that the integer part of the difference between the example phases of *Ref* and *Div* in Fig. 2.18 is different from zero. Moreover, the instants associated with the first edges of *Div* that occur after the edges of *Ref* are $t_{Div}[2]$ and $t_{Div}[5]$, respectively. The phase differences $\Phi_{diff}(t_{Div}[2])$ and $\Phi_{diff}(t_{Div}[5])$ at the instants $t_{Div}[2]$ and $t_{Div}[5]$ are equal to -0.75 and to -2.8 , respectively. The outputs out_{cntRef} and out_{cntDiv} of the counters in Fig. 2.17 are equal to the integer parts of the phases of *Ref* and *Div*, respectively, as shown in Fig. 2.18. The phase of *Ref* with integer and fractional parts is obtained by adding out_{cntRef} and out_{TDC} . The phase of *Div* is given by out_{cntDiv} . Hence, a digital measurement e of the phase difference is given by:

$$e = out_{cntRef} + out_{TDC} - out_{cntDiv}. \quad (2.7)$$

By applying Eq. (2.7) to the example in Fig. 2.18, the digitized phase differences $e(t_{Div}[2])$ and $e(t_{Div}[5])$ are equal to -0.75 and to -2.8 , respectively, as expected.

The phase-to-digital conversion realized by the TDC in the ADPLL in Fig. 2.17 is represented by the block diagram in Fig. 2.19. The output of the block labelled “Ideal Phase Measurement on the Edges of *Div*” is equal to the phase of its input on the edges of *Div*. The output of the block labelled “Modulus and Conversion to Seconds” is equal to the modulus of the input multiplied by T_{Ref} . The blocks labelled “TDC Quantization” and “Normalization by T_{Ref} ” represent the quantization in the TDC and a division by T_{Ref} , respectively.

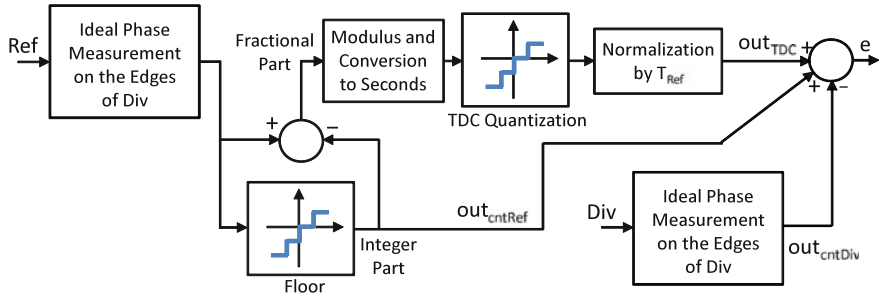


Fig. 2.19 Model of the phase digitization realized by system that comprises the block TDC and the counters in Fig. 2.17

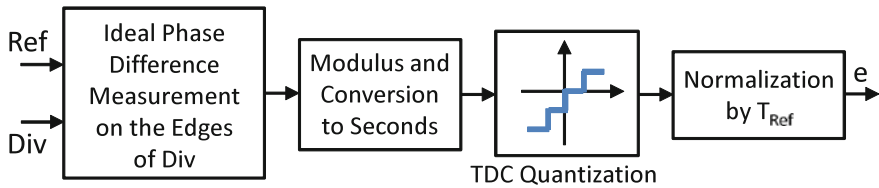


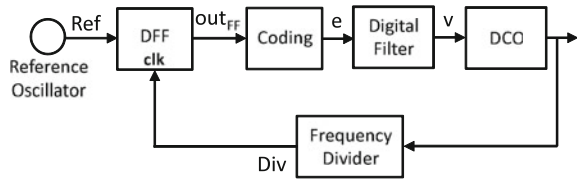
Fig. 2.20 Model of the phase digitization realized by the TDC in Fig. 2.16

2.5.2 Phase Difference with Integer Part Equal to Zero

The output of the TDC in the TDC-based ADPLL in Fig. 2.16 is a digitized version of the phase difference between *Ref* and *Div* when the integer part of the phase difference is equal to zero. Moreover, the TDC can only measure the *positive* values of the fractional part of the phase difference. Hence, it is important to introduce a fixed delay between *Ref* and *Div* in order to guarantee that the fractional part of the phase error is always positive. For example, a fixed delay between *Ref* and *Div* can be set by introducing a delay element (such as a non-inverting buffer) with delay t_{offset} at one of the TDC inputs and subtracting from the TDC output a constant which is equal to the output produced by the TDC when the time interval that has to be measured is equal to t_{offset} . In fact, an ideal TDC in a noiseless TDC-based ADPLL in this configuration produces an output equal to t_{offset} when the ADPLL has reached steady state. The input of the digital filter in this ADPLL is given by the difference between the TDC output and t_{offset} . Hence, the input of the digital filter is equal to zero and the DCO frequency remains constant. As a consequence, the phase difference between *Ref* and *Div* is constant and equal to t_{offset} .

The phase-to-digital conversion realized by the TDC in a TDC-based ADPLL is represented by the block diagram in Fig. 2.20 when the integer part of the phase difference is equal to zero. The output of the block labelled “Ideal Phase Difference Measurement on the edges of Div” is equal to the phase difference between *Ref* and *Div*. The output of the block “Ideal Phase Difference Measurement on the edges

Fig. 2.21 Block diagram of a flip-flop-based ADPLL



of Div ” is equal to the block “Ideal Phase Measurement on the edges of Div ” with input Ref in Fig. 2.19 when the integer part of the phase difference is equal to zero. The blocks “Modulus and Conversion to Seconds”, “TDC Quantization” and “Normalization by T_{Ref} ” in Fig. 2.20 realize the same operations as the blocks with the same names in Fig. 2.19.

2.6 Phase Digitization in a Flip-Flop-Based ADPLL

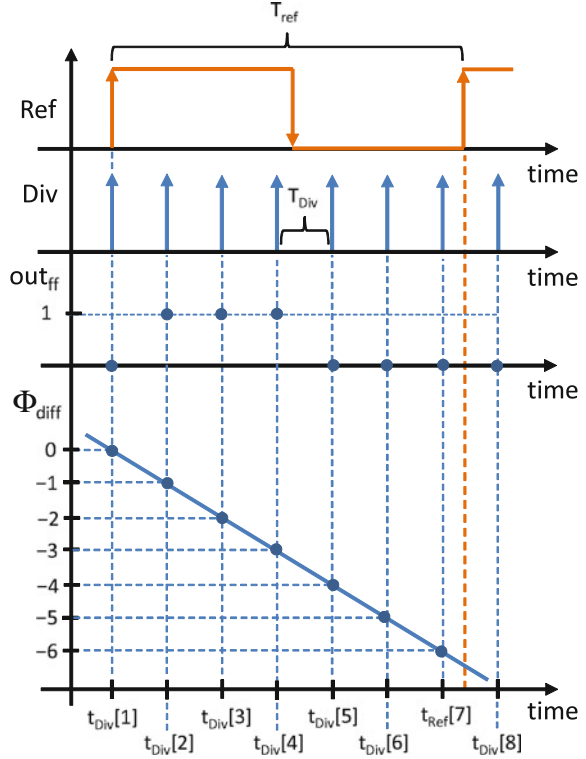
A particular case of a TDC-based ADPLL is the flip-flop based ADPLL [6, 7] shown in Fig. 2.21.

When the integer part of the phase difference is zero, the output of the flip-flop in Fig. 2.21 is equal to 1 if a rising edge of Ref occurred before that of Div ; otherwise it is zero. Hence, the output of the TDC is equal to zero and one when the phase difference on the edges of Div is negative and positive, respectively, assuming that the integer part of the phase difference is zero. When the integer part of the phase difference is different from zero, the level of the output of the flip-flop is not determined by the sign of the phase difference. For example, assume the example timing diagram in Fig. 2.22.

In Fig. 2.22, the period of Div is much smaller than that of Ref and the integer part of the phase difference is zero or negative. The output of the flip-flop is equal to one and zero when the edges of Div are included in the first and second halves of the period of Ref , respectively. Hence, the output of the flip-flop is not unique when the sign and the integer part of the phase difference are fixed and different from zero, respectively. We conclude that it is necessary that the integer part of the phase difference is always equal to zero in the flip-flop-based ADPLL shown in Fig. 2.21 in order to have a bilinear relationship between the output of the flip-flop and the sign of the phase difference.

The output of the block labelled “Coding” in Fig. 2.21 is the digital word $-wrd$ or $+wrd$ when the input is zero or one, respectively. The “Coding” block is necessary in order to increase or decrease the DCO frequency when the phase difference is positive or negative, respectively. In fact, if the output of the flip-flop in Fig. 2.22 is fed to the input of the digital filter directly, the input of the digital filter is zero when the phase difference between Ref and Div is negative. The output of a generic digital

Fig. 2.22 Example timing diagram of the edges of *Ref* and *Div*, the phase difference between *Ref* and *Div* and the output *out_{ff}* of the flip-flop in Fig. 2.21, when the integer part of the phase difference is zero or negative



filter with one pole at the origin and several zeros and poles in the left half-plane² does not change when the input is zero. Hence, the ADPLL in Fig. 2.21 does not change the frequency or phase of the DCO when the phase difference between *Ref* and *Div* is zero and the D-flip-flop is connected to the digital filter directly. By contrast, the output of the “Coding” block in Fig. 2.21 is negative when the phase difference is negative. Thus, the DCO frequency decreases when the phase difference between *Ref* and *Div* is negative. The modulus of the digital words produced by the “Coding” block is equivalent to the gain of the TDC in the TDC-based ADPLL in Fig. 2.16.

Notice that it is important to determine when the phase measurements are produced in a flip-flop-based ADPLL in order to determine how the digital filter is clocked. In the ADPLL in Fig. 2.21, the measurement of the phase difference is produced on the edges of *Div*. Hence, we assume that the digital filter in a flip-flop-based ADPLL is clocked by *Div*.

Figure 2.23 shows a behavioral model associated with the phase-to-digital conversion in a flip-flop-based ADPLL. The block labelled “Ideal Phase Difference Measurement on the Edges of *Div*” in Fig. 2.23 produces an ideal measurement of

² We assume a generic digital filter with one pole in the origin in order to have a type-II PLL. Moreover, additional zeros and poles have to be in the left hand plane to guarantee a stable loop.

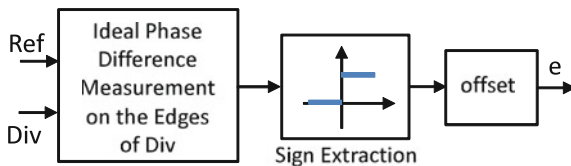


Fig. 2.23 Model of the phase-to-digital conversion associated with the flip-flop in the flip-flop-based ADPLL in Fig. 2.21 when the integer part of the phase difference is equal to zero

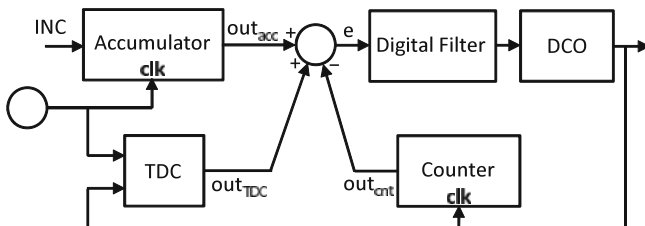


Fig. 2.24 Block diagram of an accumulator-based ADPLL

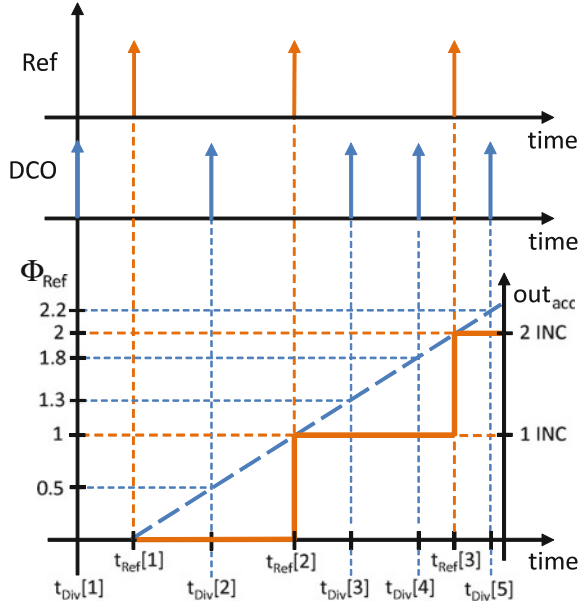
the phase difference between *Ref* and *Div* on the rising edges of *Div*. The block “Ideal Phase Difference Measurement” represents the operations of an ideal flip-flop. The output of the block labelled “Sign Extraction” in Fig. 2.23 is equal to 1 when the input is positive or zero. Otherwise, the output of the block “Sign Extraction” is zero. The main difference between a flip-flop-based and a TDC-plus-PFD based ADPLL is that a flip-flop-based ADPLL does not produce a measurement of the amplitude of the modulus of the phase difference. In fact, a flip-flop-based ADPLL discriminates only between the cases when the phase difference is positive or negative. Hence, the phase difference between *Ref* and *Div* cannot be set to a constant in an ideal noiseless flip-flop-based ADPLL at steady state. By contrast, in ideal noiseless PFD-plus-TDC-based and TDC-based ADPLLs the phase difference between *Ref* and *Div* can be a constant when the ADPLL has reached steady state. However, a flip-flop-based ADPLL requires only one flip-flop to produce the phase-to-digital conversion.

2.7 Phase Digitalization in an Accumulator-Based ADPLL

Figure 2.24 shows the block diagram of an accumulator-based ADPLL [8, 9].

The phase-to-digital conversion in the accumulator-based ADPLL in Fig. 2.24 can be studied in two cases: when the integer part of the difference between *INC* times the phase of the reference oscillator and that of the DCO is (i) different from zero and (ii) equal to zero. Notice that we will use the term “phase difference” in association with accumulator-based ADPLLs to refer to the difference between *INC*

Fig. 2.25 Phase of the reference oscillator and output of the accumulator in an accumulator-based ADPLL



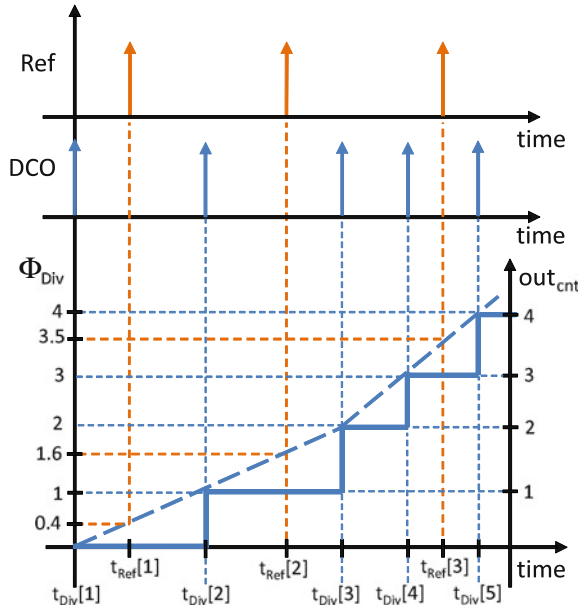
times the phase of the reference oscillator and that of the DCO. In the following, we will focus first on the operation of an accumulator-based ADPLL when the integer part of the phase difference is different from zero. Then, we will consider the case in which the integer part of the phase difference is *equal* to zero.

2.7.1 Phase Difference with Integer Part Different from Zero

Before discussing the operation of an accumulator-based ADPLL, notice that the accumulator and the counter in Fig. 2.24 operate in the clock domains of the reference oscillator and the DCO, respectively. Thus, in order to realize subtraction between the outputs of the accumulator and the counter, it is necessary to sample the output of the accumulator on the edges of the DCO or to sample the output of the counter on those of *Ref*. This sampling operation is referred to as “retiming” in most papers [8, 9]. In order to keep our analysis simple, we first discuss the basic operations of an ADPLL without retiming. Then, we will analyse the impact of retiming on the ADPLL.

First, let us assume that there is at least one edge of the DCO between two consecutive edges of the reference oscillator. This assumption is consistent with the fact that the frequency of the DCO is usually much higher than that of the reference oscillator.

Fig. 2.26 Phase of the DCO and output of the counter in an accumulator-based ADPLL



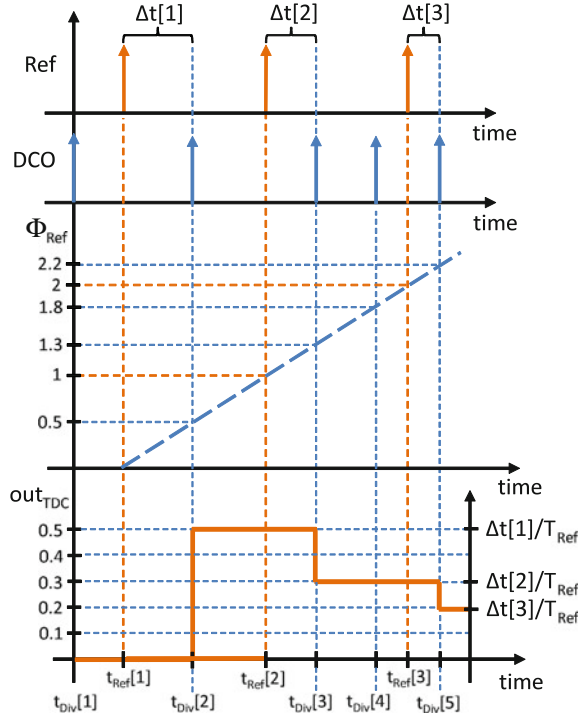
The output of the accumulator in Fig. 2.24 is equal to the integer part of the phase of Ref multiplied by INC , according to the definition of phase that we used in Sect. 2.2. Figure 2.25 shows the relationship between the output of the accumulator in Fig. 2.24 and the phase of Ref . Notice that there is no frequency divider in the accumulator-based ADPLL in Fig. 2.24. The frequency divider is not needed because the ratio between the frequencies of the reference oscillator and the DCO is set by INC . In fact, if the phase difference at steady state is zero the phase of the DCO is equal to INC times that of the reference oscillator. Phase is the integral of frequency. Hence, the frequency of the DCO at steady state is equal to INC times that of the reference oscillator.

Similar to the accumulator, the output of the counter is equal to the integer part of the phase of the DCO, as shown in Fig. 2.26.

Let us assume that the time interval measured by the TDC in Fig. 2.24 is determined by an edge of Ref and the successive DCO edge. From the definitions of phase in Sect. 2.2, we deduce that the time interval measured by the TDC in Fig. 2.24 divided by T_{Ref} is a measure of the *fractional part* of the phase of the *reference oscillator* at the instants of the DCO edges, as illustrated in Fig. 2.27. Hence, the output of the TDC has to be *added* to the output of the accumulator. Notice that if the TDC is reset and started on the edges of the DCO and stopped on those of Ref , out_{TDC} is related to the fractional part of the *phase of the DCO* and the normalization factor of the TDC has to be the period of the DCO .

The phase difference in Fig. 2.24 is produced by a digital adder with three inputs, out_{acc} , out_{TDC} and out_{cnt} . The inputs to the digital adder are associated with *two*

Fig. 2.27 Phase of the reference oscillator and output of the TDC in an accumulator-based ADPLL



different time domains. In fact, out_{acc} is associated with the time domain of the edges of Ref . By contrast, the signals out_{TDC} and out_{cnt} are related to the time domain of the edges of the DCO. In order to produce the addition in a single time domain, the inputs of the digital adder have to be sampled. By retiming the reference oscillator in an accumulator-based ADPLL, it is possible to have the signals out_{acc} , out_{TDC} and out_{cnt} in the same time domain.

Figure 2.28, shows an accumulator-based ADPLL with retiming [9]. The blocks labelled “DFF” in Fig. 2.28 which are connected to the outputs of the counter and the TDC represent arrays of flip-flops that sample the bits of the counter and TDC outputs, respectively. Retiming does not change the fact that the outputs of the accumulator and the counter in Fig. 2.28 are equal to the integer parts of the phases of the reference oscillator and the DCO on the edges of the output Ref_R of the block “Retiming”. In fact, the output of the accumulator in Fig. 2.28 on the edges of Ref_R is equal to the output of an accumulator clocked by Ref and sampled on the edges of Ref_R . The TDC in Fig. 2.28 measures the fractional part of the phase of the reference oscillator on the edges of Ref_R . The TDC in Fig. 2.28 is started and stopped on the edges of Ref and the next edges of the DCO, respectively. Hence, the normalization factor of the TDC output is the period of the reference oscillator. Different from a PFD-plus-TDC based ADPLL, the normalization factor of the TDC in an accumulator-based ADPLL with retiming does not have to be changed over time because the TDC is

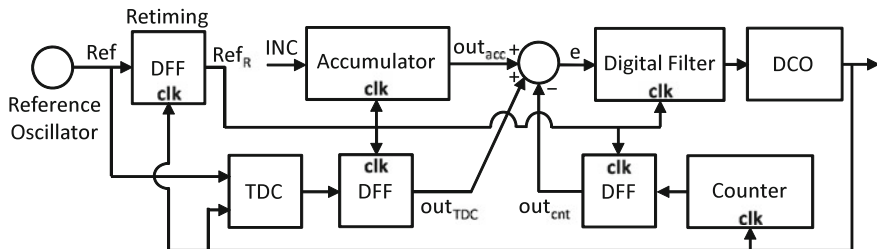


Fig. 2.28 Block diagram of an accumulator-based ADPLL with retiming based on sampling the output of the reference oscillator on the edges of the DCO

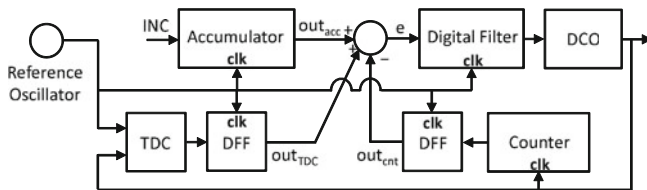


Fig. 2.29 Block diagram of an accumulator-based ADPLL with retiming based on the edges of the reference oscillator

always stopped on the edges of the same signal. Notice that the output of the TDC has to be multiplied by INC in order to have the measurement of the fractional part of the phase of Ref consistent with the measurement of the integer part produced by the accumulator.

The loop filter in Fig. 2.28 is clocked by the output Ref_R of the block “Retiming”. The signal Ref_R has a variable period because it is obtained by sampling the output reference oscillator with the DCO output. In Sect. 2.10.2, we will discuss the effect of a variable period of the clock of the digital filter.

An alternative approach to retiming in Fig. 2.28 is shown in Fig. 2.29 [10]. In the approach in Fig. 2.29, the clock of the digital filter has a constant period and issues related to a noisy clock are thereby avoided in the digital filter [10]. The TDC in Fig. 2.29 is reset and started on the edges of the DCO and stopped on those of the reference oscillator [10].

The phase-to-digital conversion in an accumulator-based ADPLL with retiming in Fig. 2.28 is represented by the block diagram in Fig. 2.30.

The outputs of the blocks labelled “Ideal Phase Measurement on the Edges of Ref_R ” in Fig. 2.30 are equal to the phases of their respective input signals measured on the edges of Ref_R . The output of the blocks labelled “Scaling by INC ” in Fig. 2.30 are equal to its input times INC . The output of the block “Modulus and Conversion to Seconds” in Fig. 2.30 is equal to the modulus of its input times T_{Ref} . The block “Normalization by T_{Ref} ” in Fig. 2.30 divides its input by T_{Ref} . The operations of the blocks labelled “Floor” and “TDC Quantization” in Fig. 2.30 are equal to those of the blocks with the same names in Fig. 2.17.

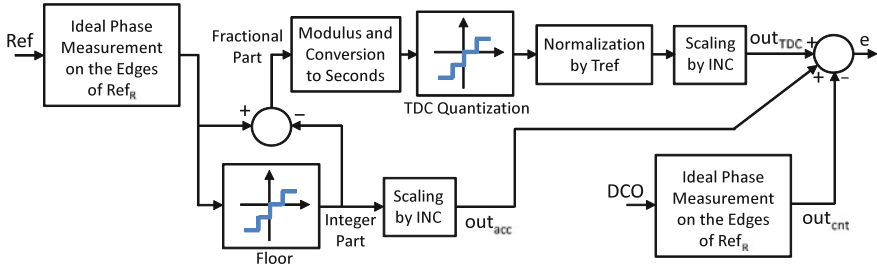


Fig. 2.30 Block diagram of an the phase-to-digital conversion in an accumulator-based ADPLL in Fig. 2.28

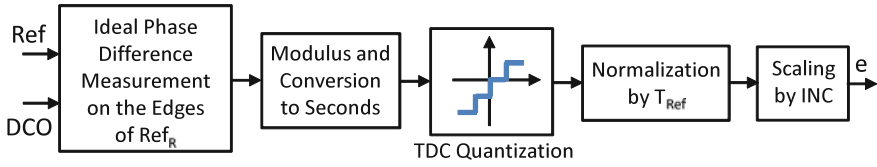


Fig. 2.31 Block diagram of the phase-to-digital conversion in an accumulator-based ADPLL in Fig. 2.28 when the integer part of the phase difference is equal to zero

The phase of the DCO in Fig. 2.28 on each edge of Ref_R is an integer. Hence, the output out_{cnt} of the counter in Fig. 2.28 on an edge of Ref_R is equal to the corresponding phase of the DCO. Thus, the output of the block “Ideal Phase Measurement on the Edges of Ref_R ” associated with the DCO in Fig. 2.30 is equal to out_{cnt} .

In an accumulator-based ADPLL, the fractional part of the phase of Ref is quantized by the TDC. By contrast, in a PFD-plus-TDC-based ADPLL, the phase difference between Ref and Div with *both integer and fractional* parts is quantized by the TDC when the integer part of the phase difference is different from zero. This is the main difference between the phase-to-difference conversion in a PFD-plus-TDC-based and an accumulator-based ADPLL.

2.7.2 Phase Difference with Integer Part Equal to Zero

The integer part of the phase difference in the accumulator based ADPLL in Fig. 2.29 is equal to zero when the number of the DCO’s edges included between two consecutive edges of the reference oscillator is equal to INC .

Assume that the time interval between the edges of Ref and the closest DCO edges is very small. Hence, the fractional part of the phase difference is not zero. Moreover, the number of complete DCO periods included between two consecutive edges of Ref is equal to $(INC - 1)$. Hence, on every edge of Ref_R in Fig. 2.29, the outputs of the

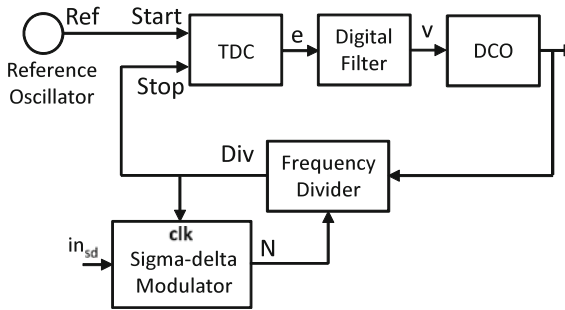


Fig. 2.32 Fractional-N TDC-based ADPLL with a sigma-delta modulator that sets the division ratio N

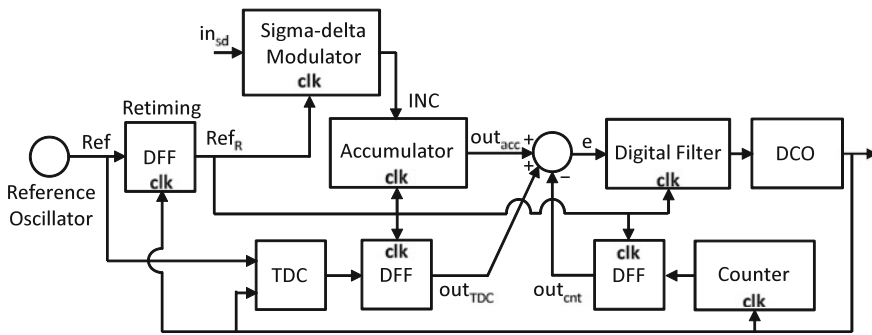


Fig. 2.33 Accumulator-based ADPLL with a sigma-delta modulator that sets the division ratio INC

counter and the accumulator are equal when the integer part of the phase difference is zero. By contrast, the fractional part of the phase difference is equal to the fractional part of the phase of Ref , i.e. the output of the TDC in Fig. 2.29. We conclude that the block diagram in Fig. 2.30 can be simplified to that in Fig. 2.31 when the integer part of the phase difference is zero in an accumulator-based ADPLL.

2.8 Fractional-N Operations

In traditional fractional-N PLLs, a sigma-delta modulator is used to set the division ratio of the frequency divider. Similar to a traditional PLL, a TDC-based ADPLL can include a sigma-delta modulator whose output determines the division ratio N , as shown in Fig. 2.32.

A similar approach can be used in accumulator-based ADPLLs.

Figure 2.33 shows an accumulator-based ADPLL with a sigma-delta modulator that sets the value of INC . To our knowledge, the architecture in Fig. 2.33 is miss-

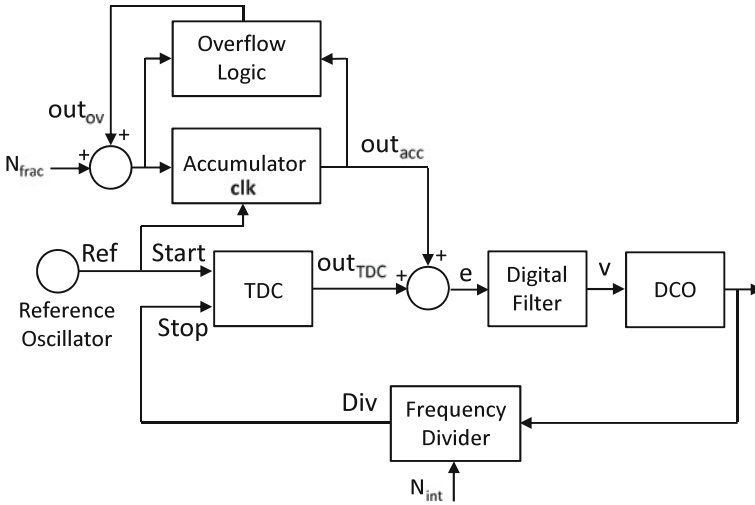


Fig. 2.34 Fractional-N TDC-based ADPLL with an accumulator that sets the fractional part N_{frac} of the division ratio

ing in the literature. By contrast, an accumulator-based ADPLL which synthesizes fractional division ratios is obtained by introducing a fractional part in the digital word INC such that the accumulator in the accumulator-based ADPLL accumulates $INC = INC_{int} + INC_{frac}$.

A similar approach can be applied to a TDC-based ADPLL by introducing an accumulator that increases by INC_{int} every time an edge of Ref occurs, as shown in Fig. 2.34.

The accumulator in Fig. 2.34 determines the fractional part of the phase of Div in the same way that the accumulator in an accumulator-based ADPLL determines the fractional and integer parts of the phase of the DCO . Hence, the output of the accumulator in Fig. 2.34 is a digital word that represents a fractional number. The block labelled “Overflow Logic” in Fig. 2.34 is necessary to guarantee that the output of the accumulator is always a fractional number. In fact, the “Overflow Logic” block reads the output $out_{acc}[n]$ of the accumulator at the n -th edge of the reference oscillator. When $INC_{frac} + out_{acc}[n]$ is greater than 1, the output of $out_{ov}[n + 1]$ is equal to -1 , otherwise $out_{ov}[n + 1]$ is zero. The architecture in Fig. 2.34 is also missing in the literature.

2.9 Comparison

In the next sections, we compare PFD-plus-TDC-based, TDC-based, flip-flop-based and accumulator-based ADPLLs in terms of phase-to-digital conversion, metastability and dynamic range of the TDC.

2.9.1 Phase-to-Digital Conversion

We recall that in Sect. 2.3, we classified the architectures of ADPLLs as “phase difference digitizing” or “phase digitizing”. By comparing Figs. 2.15, 2.20 and 2.31, we deduce that PFD-plus-TDC-based, TDC-based and accumulator-based ADPLLs are described by equivalent phase-to-digital models when the integer part of the phase difference is zero. In fact, PFD-plus-TDC-based, TDC-based and accumulator-based ADPLLs can be classified as “phase difference digitizing” architectures when the integer part of the phase difference is zero.

By contrast, TDC-based ADPLLs with counters and accumulator-based ADPLLs can be considered as “phase digitizing” ADPLLs when the integer part of the phase difference is not zero, as can be deduced from the models in Figs. 2.19 and 2.30.

2.9.2 Metastability

Errors in the phase-to-digital conversion can be caused by metastability. Hence, it is important to compare the main ADPLL architectures in terms of the effects of metastability on the phase-to-digital conversion.

Metastability occurs in a flip-flop triggered by a clock when the instant at which the clock triggers the flip-flop is too close to the instant at which the input of the flip-flop changes logic level. Metastability can cause a D-flip-flop that is used as a sampler to produce sampling errors when the input and the clock are associated with two different clock domains. In PFD-plus-TDC-based, TDC-based and flip-flop-based ADPLLs, metastability can occur in the PFD, the TDC and the flip-flop. By contrast, in an accumulator-based ADPLL with retiming, metastability can occur in the TDC, the retiming block and the sampler at the output of the counter.

In a PFD with inputs *Ref* and *Div*, metastability can occur when the edges of *Ref* and *Div* are too close, namely when the phase difference is equal to zero. If the ADPLL is designed to have a non-zero phase difference at steady state, metastability issues in the PFD can be avoided. In a TDC, the effect of metastability on the output depends on the TDC architecture. In order to keep our analysis simple, we consider a delay-line-based TDC as an example architecture.

Figure 2.35 shows the block diagram of a delay-line-based TDC in which the phase-to-digital conversion is realized by propagating the start signal through a delay line and by sampling the outputs of the delays with D-flip-flops. Assume that all the delay elements in Fig. 2.35 are characterized by the same delay res . Hence, res is the *time resolution* of the TDC. Let us assume that N_{met} is the number of delay elements whose inputs transitions are close enough to the edge of *Stop* to cause metastability. Hence, the maximum difference between the time interval that has to be measured and the TDC output is equal to N_{met} times res . The maximum difference between the ideal and the digitized phase-difference caused by metastability in PFD-plus-TDC-based,

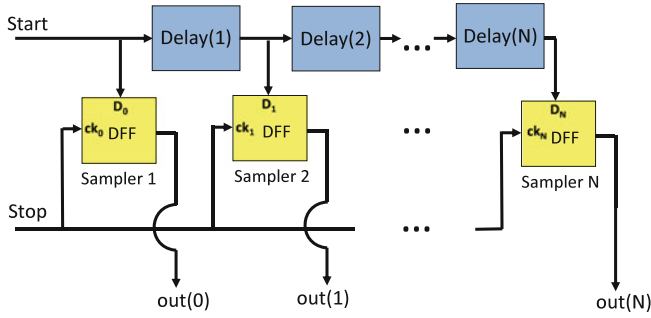


Fig. 2.35 Block diagram of a delay-line-based TDC

TDC-based and accumulator-based ADPLLs with a delay-line based TDC is equal to $res \cdot N_{met} / T_{Ref}$.

Metastability in the retiming flip-flop in Fig. 2.28 can produce a measurement of phase difference equal to minus one cycle of the reference clock when the actual phase difference is zero. In fact, one edge of Ref is always associated with one edge of the output Ref_R of the retiming flip-flop in Fig. 2.28, when the period of the DCO is much smaller than that of the reference oscillator. However, because of metastability in the retiming flip-flop, there can be an edge of Ref which is not associated with an edge of Ref_R . When there is one edge of Ref without a corresponding edge of Ref_R , an incorrect phase difference measurement equal to minus one cycle of the reference clock is produced.

Metastability in the block “D-flip-flop” at the output of the counter in Fig. 2.28 can produce a phase difference measurement included in the interval $[0, 2^b]$ DCO cycles, where b is the number of bits in the counter, when the actual phase difference is zero. In fact, we recall that the block “D-flip-flop” which samples the output of the counter in Fig. 2.28 represents an array of flip-flops that sample the counter output. Metastability in the flip-flops that sample the output of the counter in Fig. 2.28 can produce a difference in the interval $[0, 2^b]$ between the output word of the counter and that of the sampling flip-flops.

Assume that the output of the counter is associated with zero phase difference. Hence, a difference in the interval $[0, 2^b]$ between the sampled and the actual output word of the counter corresponds to a phase difference measurement included in the interval $[0, 2^b]$ DCO cycles when the actual phase difference is zero. Notice that $[0, 2^b]$ DCO cycles are approximately equal to $[0, 2^b / INC]$ cycles of the reference oscillator when the ADPLL has reached steady state. Therefore, metastability on the flip-flops that sample the counter output in an accumulator-based ADPLL has a much stronger impact on the difference between digitized and actual phase difference than metastability in the TDC in accumulator-based, PFD-plus-TDC and TDC-based ADPLLs.

2.9.3 TDC Dynamic Range

The dynamic range of a TDC is determined by the maximum duration of the time interval that the TDC can measure. The value of the dynamic range that is required by a TDC in an ADPLL depends on the ADPLL operation in terms of (i) maximum phase difference caused by noise and (ii) ratio between the frequencies of the reference oscillator and the DCO.

The maximum phase difference caused by noise in an ADPLL is related to the ADPLL noise transfer function, as shown in Chap. 4. In Chap. 4, we will show that the noise transfer functions of a TDC-based and an accumulator-based ADPLL are very similar when the integer part of the phase difference is zero. Assume that a TDC-based and an accumulator-based ADPLL are associated with the same noise transfer function. Hence, the maximum phase difference caused by noise is the same in the two ADPLL when the noise sources in the ADPLLs are the same and the integer part of the phase difference is zero. We conclude that TDC-based and accumulator-based ADPLLs require the same TDC dynamic range when they are associated with the same transfer function and the integer part of the phase difference is zero.

In Sect. 2.8, we have shown that a TDC-based and an accumulator-based ADPLL can include a sigma-delta modulator that determines N and INC , respectively, in order to synthesize a fractional ratio between the frequencies of the reference oscillator and the DCO. In these fractional- N TDC-based and accumulator-based ADPLLs, the maximum variations in N and INC , respectively, determine the required dynamic range of the TDC. The variations in N and INC are related to the output of the sigma-delta modulator. It is important to notice that the maximum variations in N and INC determine the variations in the phase difference with an integer part *different from zero*. The models that we have shown in Sects. 2.5.1 and 2.7.1, can be used to determine the maximum phase differences associated with the maximum variations in N and INC in a TDC-based and an accumulator-based ADPLL, respectively. The required TDC dynamic range is the same in a TDC-based and an accumulator-based ADPLL in which the maximum phase difference associated with the maximum variation in N and INC is the same.

2.10 Additional Design Considerations

2.10.1 Introducing Digital-Sigma-Delta-Modulators in the Architectures

A digital sigma-delta modulator is often introduced between the digital filter and the DCO in all of the ADPLL architectures that we have considered. A digital sigma-delta modulator between the digital filter and the DCO does not change how the phase-to-digital conversion is realized in the ADPLL architectures that we have analysed.

A sigma-delta modulator followed by a DCO can be considered as an equivalent DCO with higher effective resolution.

TDCs with n th-order noise-shaping of the quantization error can be used in ADPLLs. An n th-order noise-shaping TDC is represented by an n th-order sigma-delta modulator in the phase-to-digital model. An n th-order noise-shaping TDC followed by a low pass filter can be considered as an equivalent TDC with higher effective resolution, as will be discussed in Chap. 3.

2.10.2 Selecting the Clock of the Digital Filter

The clock of the digital filter in an ADPLL has to guarantee that the digital filter samples and processes every measurement that is produced by the TDC.

When the integer part of the phase difference between Ref and Div or between Ref and Ref_R is equal to zero, the TDC in a TDC-based ADPLL or in an accumulator-based ADPLL produces one measurement of phase difference per cycle of the reference oscillator. Notice that only one cycle of Div or Ref_R is associated with a cycle of Ref when the integer part of the phase difference between Ref and Div or between Ref and Ref_R is equal to zero. Hence, in a TDC-based ADPLL, a digital filter that is clocked by Ref or Div samples and processes each measurement produced by the TDC when the integer part of the phase difference is zero. In an accumulator-based ADPLL, a digital filter clocked by Ref or Ref_R samples and processes each measurement produced by the TDC when the integer part of the phase difference is zero.

However, it is important to notice that the edges of Div and Ref_R are determined by those of the DCO. By contrast, the edges of Ref are determined by a crystal oscillator in ADPLLs that are used as frequency synthesizers. The noise associated with the edges of the reference signal is typically much lower than that in the DCO when the reference signal is provided by a crystal oscillator. A noisy clock can compromise the operation of the components in the digital filter. Hence, issues related to a noisy clock can be avoided by clocking the digital filter with the reference oscillator.

2.11 Conclusions

In this chapter, we have reviewed 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 have derived 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 have discussed possible strategies to clock the digital filter in the various ADPLL architectures. We have also showed how to modify the ADPLL architectures to synthesize a fractional ratio between the frequencies of the reference oscillator and the DCO. Finally, we have compared the ADPLL architectures in terms of phase-to-digital conversion, TDC dynamic range, and metastability.

References

1. 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.
2. C.-W. Yao, A. N. Willson, "A 2.8–3.2-GHz Fractional-N Digital PLL with ADC-Assisted TDC and Inductively Coupled Fine-Tuning DCO," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 698–710, Mar. 2013.
3. V. Kratyuk, P. K. Hanumolu, U.-K. Moon, K. Mayaram, "A Design Procedure for All-Digital Phase-Locked Loops Based on a Charge-Pump Phase-Locked-Loop Analogy" *IEEE Trans. Circuits Sys. II: Express Briefs*, vol. 44, no. 3, pp. 247–251, Mar. 2007.
4. K. Nagaraj, A. S. Kamath, K. Subburaj, G. Chattopadhyay, G. Nayak, S. S. Evani, N. P. Nayak, I. Prathapan, F. Zhang, B. Haroun, "Architectures and Circuit Techniques for Multi-Purpose Digital Phase Lock Loops", *IEEE Trans. Circuits Sys. I: Regular Papers*, vol. 60, no. 3, pp. 517–528, Apr. 2013.
5. 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.
6. 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.
7. N. Da Dalt, "Theory and Implementation of Digital Bang-Bang Frequency Synthesizers for High Speed Serial Data Communications," *PhD Thesis*, RWTH Aachen University, Feb. 2007.
8. 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.
9. R. B. Staszewski, P. T. Balsara, "All-Digital Frequency Synthesizer in Deep-Submicron CMOS," John Wiley & Sons Inc., 2006.
10. S. Mendel, C. Vogel, N. Da Dalt, "A Phase-Domain All-Digital Phase-Locked Loop Architecture Without Reference Clock Retiming," *IEEE Trans. Circuits Sys. II: Express Briefs*, vol. 56, no. 11, pp. 860–264, Nov. 2009.

Noise-Shaping All-Digital Phase-Locked Loops

Modeling, Simulation, Analysis and Design

Brandonisio, F.; Kennedy, M.P.

2014, XIII, 177 p. 145 illus., 79 illus. in color., Hardcover

ISBN: 978-3-319-03658-8