Design of Monolithic Phase-Locked Loops and Clock Recovery Circuits—A Tutorial

Behzad Razavi

Abstract — **This paper describes the principles of phase-locked system design with emphasis on monolithic implementations. Following a brief review of basic concepts, we analyze the static and dynamic behavior of phase-locked loops and study the design of their building blocks in bipolar and CMOS technologies. Next, we describe charge-pump phase-locked loops, effect of noise, and the problem of clock recovery from random data. Finally, we present applications in communications, digital systems, and RF transceivers.**

1. INTRODUCTION

PHASE-LOCKED loops (PLLs) and clock recovery circuits (CRCs) find wide application in areas such as communications, wireless systems, digital circuits, and disk drive electronics. While the concept of phase locking has been in use for more than half a century, monolithic implementation of PLLs and CRCs has become possible only in the last twenty years and popular in the last ten years. Two factors account for this trend: the demand for higher performance and lower cost in electronic systems, and the advance of integrated-circuit (IC) technologies in terms of speed and complexity.

This tutorial deals with the analysis and design of monolithic PLLs and CRCs. Following a brief look in Section 2 at a number of design problems that can be solved using phase locking, we review some basic concepts in Section 3 to establish proper background as well as define the terminology. In Section 4, we introduce the phase-locked loop in a simple form, analyze its static and dynamic behavior, and formulate its limitations. In Section 5, we describe the design of circuit building blocks, and in Section 6, charge-pump PLLs. Sections 7 to 9 deal with phase noise, clock recovery circuits, and applications of phase-locked systems, respectively.

2. WHY PHASE-LOCK?

Phase locking is a powerful technique that can provide elegant solutions in many applications. In this section, we consider four design problems that can be efficiently solved with the aid of PLLs. We return to these problems after studying the principles of phase locking.

2.1 Jitter Reduction

Signals often experience timing jitter as they travel through a communication channel or as they are retrieved from a storage medium. Depicted in Figure 1, jitter manifests itself as variation of the period of a waveform, a type of corruption that cannot be removed by amplification and clipping even if the signal is binary.

A PLL can be used to reduce the jitter.

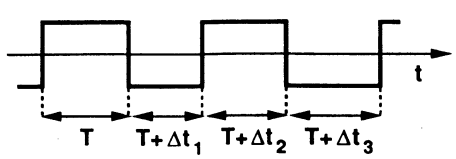

Fig. 1 Timing jitter.

2.2 Skew Suppression

Figure 2 illustrates a critical problem in high-speed digital systems. Here, a system clock, CK_S, enters a chip from a printed-circuit (PC) board and is buffered (in several stages) to sharpen its edges and drive the load capacitance with minimal delay. The principal difficulty in such an arrangement is that the on-chip clock, CK_C, typically drives several nanofarads of device and interconnect capacitance, exhibiting significant delay with respect to CK_S. The resulting skew reduces the timing budget for on-chip and inter-chip operations.

In order to lower the skew, the clock buffer can be placed in a phase-locked loop, thereby aligning CK_C with CK_S.

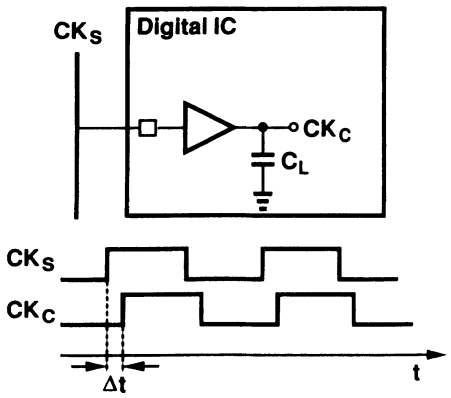

Fig. 2 Clock skew in a digital system.

2.3 *Frequency Synthesis*

Many applications require frequency multiplication of periodic signals. For example, in the digital system of Figure 2, the bandwidth limitation of PC boards constrains the frequency of CK_S, whereas the on-chip clock frequency may need to be much higher. As another example, wireless transceivers employ a local oscillator whose output frequency must be varied in small, precise steps, for example, from 900 MHz to 925 MHz in steps of 200 kHz.

These exemplify the problem of "frequency synthesis," a task performed efficiently using phase-locked systems.

2.4 *Clock Recovery*

In many systems, data is transmitted or retrieved without any additional timing reference. In optical communications, for example, a stream of data flows over a single fiber with no accompanying clock, but the receiver must eventually process the data *synchronously*. Thus, the timing information (e.g., the clock) must be recovered from the data at the receive end (Fig. 3). Most clock recovery circuits employ phase locking.

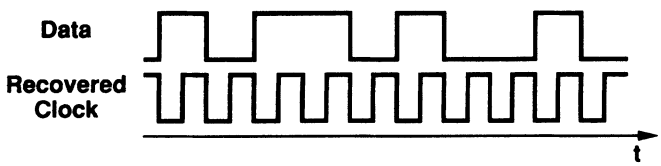

Fig. 3 Clock recovery from random data.

3. Basic Concepts

In studying PLLs and CRCs, we often need to draw upon concepts from the theory of signals and systems. This section provides a brief review of such concepts to the extent that they prove useful to IC designers.

3.1 *Time- and Frequency-Domain Characteristics*

Phase-locked systems exhibit nonlinear behavior at least during part of their operation (e.g., transients), thus requiring time-domain analysis in almost all applications. However, in the steady state and during slow transients, it is extremely helpful to study the response in the frequency domain as well, especially if the application imposes certain constraints on the purity of the output spectrum.

Most of the signals encountered in PLLs are either strictly periodic, for example, $x(t) = A\cos\omega_c t$, or phase-modulated, for example,

$$x(t) = A\cos[\omega_c t + \phi_n(t)] \tag{1}$$

We consider the second example as a more general case. The total phase of this signal is defined as $\phi_c(t) = \omega_c t + \phi_n(t)$ and the total frequency as $\Omega_c(t) = d\phi_c/dt = \omega_c + d[\phi_n(t)]/dt$.[1] PLLs usually operate on the "excess" components of ϕ_c and Ω_C, that is, $\phi_n(t)$ and $d[\phi_n(t)]/dt$, respectively.

In most cases of interest, $|\phi_n(t)| \ll 1$ radian, i.e., the difference between consecutive periods of the waveform is small and the signal only slightly deviates from a strictly periodic behavior. We call such a signal "almost periodic." This is illustrated in Figure 4, where two waveforms with equal "average" frequencies are shown. Note that while each period of the waveform in Figure 4b is very close to that in Figure 4a, the *phase difference* between the two can grow significantly because small phase deviations accumulate after every period.

[1]The quantity Ω_c is actually called the *angular frequency* and measured in rad/s, but for the sake of brevity we simply call it the *frequency*.

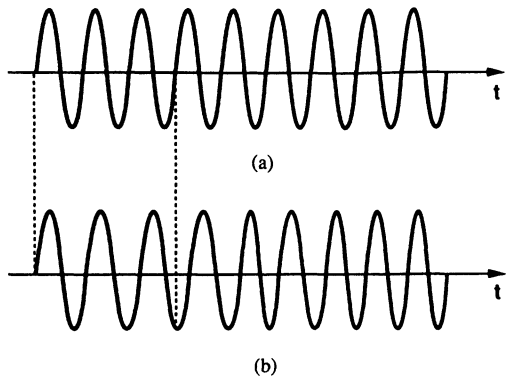

Fig. 4 (a) Periodic and (b) almost-periodic waveforms.

Let us consider examples of $\phi_n(t)$ in Eq. (1) that lead to the behavior depicted in Figure 4b. First, suppose $\phi_n(t) = \phi_m \sin \omega_m t$, where $\phi_m \ll 1$ radian. Then, (1) can be simplified as

$$x_1(t) = A \cos \omega_c t \cos(\phi_m \sin \omega_m t) - A \sin \omega_c t \sin(\phi_m \sin \omega_m t) \tag{2}$$

$$\approx A \cos \omega_c t - (A \sin \omega_c t)(\phi_m \sin \omega_m t) \tag{3}$$

$$= A \cos \omega_c t + \frac{A\phi_m}{2}[\cos(\omega_c + \omega_m)t - \cos(\omega_c - \omega_m)t] \tag{4}$$

Thus, the waveform has a strong component at $\omega = \omega_c$ and two small "sidebands" at $\omega = \omega_c \pm \omega_m$ (Fig. 5a). Second, suppose $\phi_n(t)$ is a stationary Gaussian random noise with a "low-pass" power spectral density:

$$P_\phi(\omega) = \frac{1}{1 + (\omega/\omega_0)^2} \tag{5}$$

Then, in a similar fashion, if $|\phi_n| \ll 1$, Eq. (1) can be expressed as

$$x_2(t) \approx A \cos \omega_c t + A\phi_n(t) \sin \omega_c(t) \tag{6}$$

The resulting spectrum exhibits noise "skirts" around the center frequency (Fig. 5b).

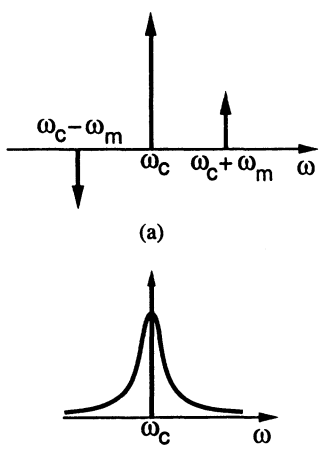

Fig. 5 (a) Sidebands and (b) noise skirts resulting from phase modulation.

From the above discussion, we can define a number of important parameters. "Cycle-to-cycle" jitter is the difference between every two consecutive periods of an almost-periodic waveform, and "absolute" jitter is the phase difference between the same waveform and a periodic signal having the same average frequency. If random, jitter is usually specified in terms of its root mean square (rms) and peak-to-peak values.

Frequency-domain counterparts of jitter are sidebands and "phase noise." As illustrated in Figure 5a, sidebands are deterministic components that do not have a harmonic relationship with the main component ω_c (also called the "carrier"); in most cases of interest, they are quite close to ω_c, i.e., $\omega_m \ll \omega_c$. Sidebands are specified with their frequency and their magnitude relative to that of the carrier.

In contrast to sidebands, phase noise arises from random frequency components (Fig. 5b). To quantify phase noise, we consider a unit bandwidth at a frequency offset $\Delta\omega$ with respect to ω_c, calculate the total noise power in this bandwidth, and divide the result by the power of the carrier (Fig. 6). Phase noise is expressed in terms of dBc/Hz, the letter "c" indicating

the normalization of the noise power to the carrier power, and the unit Hz signifying the unity bandwidth used for the noise power.

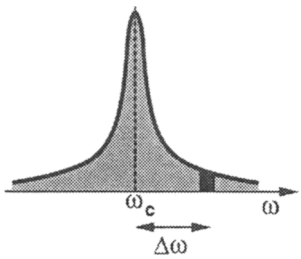

Fig. 6 Phase noise measurement.

It is interesting to note that although the phase modulation of $x_1(t)$ is deterministic and that of $x_2(t)$ random, their time-domain jitter may appear to be the same. In other words, if uniformity of zero crossings is critical, sidebands are as undesirable as random noise.

3.2 Voltage-Controlled Oscillator

An ideal voltage-controlled oscillator (VCO) generates a periodic output whose frequency is a linear function of a control voltage v_{cont}:

$$\omega_{out} = \omega_{FR} + K_{VCO} V_{cont} \tag{7}$$

where ω_{FR} is the "free-running" frequency and K_{VCO} is the "gain" of the VCO (specified in rad/s/V). Since phase is the time integral of frequency, the output of a sinusoidal VCO can be expressed as

$$y(t) = A \cos\left(\omega_{FR} t + K_{VCO} \int_{-\infty}^{t} V_{cont} dt\right) \tag{8}$$

In practical VCOs, K_{VCO} exhibits some dependence on the control voltage and eventually drops to zero as $|V_{cont}|$ increases. This is explained in Section 4.3.

It is interesting to note that if $V_{cont}(t) = V_m \cos \omega_m t$, then

$$y(t) = A \cos\left(\omega_{FR} t + \frac{K_{VCO}}{\omega_m} V_m \sin \omega_m t\right) \tag{9}$$

Called the *modulation index*, the quantity K_{VCO}/ω_m decreases as the modulating frequency ω_m increases, i.e., the VCO has a natural tendency to reject high-frequency components applied at its control input.

In studying PLLs, we usually consider a VCO as a linear time-invariant system, with the control voltage as the system's input and the *excess phase* of the output signal as the system's output. Since the excess phase,

$$\phi_{out}(t) = K_{VCO} \int V_{cont} dt \tag{10}$$

the input/output transfer function is

$$\frac{\Phi_{out}(s)}{V_{cont}(s)} = \frac{K_{VCO}}{s} \tag{11}$$

Equation (10) reveals an interesting property of VCOs: to change the output *phase*, we must first change the *frequency* and let the integration take place.[2] For example, suppose for $t < t_0$, a VCO oscillates at the same frequency as a reference but with a finite phase error (Fig. 7). To reduce the error, the control voltage, V_{cont}, is stepped by $+\Delta V$ at $t = t_0$, thereby increasing the VCO frequency and allowing the output to accumulate phase faster than the reference. At $t = t_1$, when the phase error has decreased to zero, V_{cont} returns to its initial value. Now, the two signals have equal frequencies and zero phase difference. Note also that the same goal can be accomplished by *lowering* the VCO frequency during this interval.

The above observation leads to another interesting result as well: the output phase of a VCO cannot be determined only from the *present* value of the control voltage, i.e., it depends on the history of V_{cont}. For this reason, we treat the output phase of VCOs as an independent initial condition (or state variable) in the time-domain analysis of PLLs.

In some cases, it is advantageous to control the output frequency of an oscillator by a *current*. Called a *current-controlled oscillator* (CCO), such a circuit exhibits the same behavior as a VCO.

[2]We assume the VCO has no other input to set its phase.

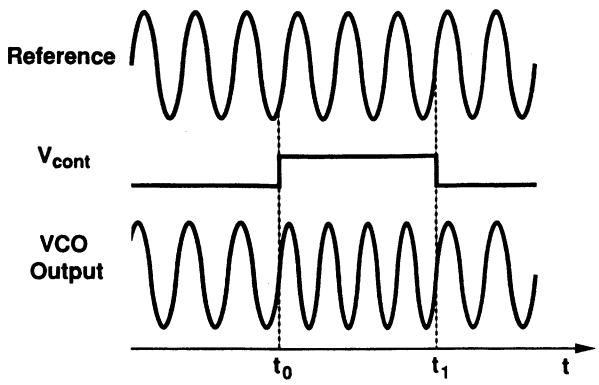

Fig. 7 Phase alignment of a VCO with a reference.

3.3 *Phase Detector*

An ideal phase detector (PD) produces an output signal whose dc value is linearly proportional to the difference between the phases of two periodic inputs (Fig. 8):

$$\overline{v_{out}} = K_{PD}\Delta\phi \tag{12}$$

where K_{PD} is the "gain" of the phase detector (specified in V/rad), and $\Delta\phi$ is the input phase difference. In practice, the characteristic may not be linear or even monotonic for large $\Delta\phi$. Furthermore, K_{PD} may depend on the amplitude or duty cycle of the inputs. These points are explained later.

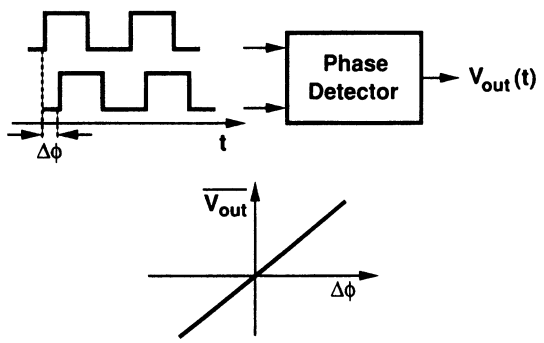

Fig. 8 Characteristic of an ideal phase detector.

Figure 9 illustrates a typical example, where the PD generates an output pulse whose width is equal to the time difference between consecutive zero crossings of the two inputs. Because the two frequencies are not equal, the phase difference exhibits a "beat" behavior with an average value of zero.

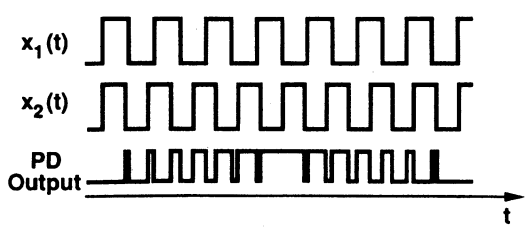

Fig. 9 Input and output waveforms of a PD.

A commonly used type of phase detector is a multiplier (also called a mixer or a sinusoidal PD). For two signals $x_1(t) = A_1 \cos\omega_1 t$ and $x_2(t) = A_2 \cos(\omega_2 t + \Delta\phi)$, a multiplier generates

$$y(t) = \alpha A_1 \cos\omega_1 t \cdot A_2 \cos(\omega_2 t + \Delta\phi) \tag{13}$$

$$= \frac{\alpha A_1 A_2}{2}\cos[(\omega_1 + \omega_2)t + \Delta\phi] + \frac{\alpha A_1 A_2}{2}\cos[(\omega_1 - \omega_2)t - \Delta\phi] \tag{14}$$

where α is a proportionality constant. Thus, for $\omega_1 = \omega_2$, the phase/voltage characteristic is given by

$$\overline{y(t)} = \frac{\alpha A_1 A_2}{2}\cos\Delta\phi \tag{15}$$

Plotted in Figure 10, this function exhibits a variable slope and nonmonotonicity, but it resembles that in Eq. (12) if $\Delta\phi$ is in the vicinity of $\pi/2$:

$$\overline{y(t)} \approx \frac{\alpha A_1 A_2}{2}\left(\frac{\pi}{2} - \Delta\phi\right) \tag{16}$$

yielding $K_{PD} = -\alpha A_1 A_2/2$. Note that the average output is zero if $\omega_1 \neq \omega_2$.

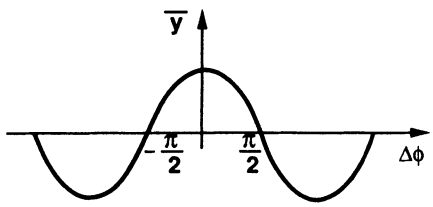

Fig. 10 Characteristic of a sinusoidal PD.

4. PHASE-LOCKED LOOP

4.1 Basic Topology

A phase-locked loop is a feedback system that operates on the *excess phase* of nominally periodic signals. This is in contrast to familiar feedback circuits where voltage and current amplitudes and their rate of change are of interest. Shown in Figure 11 is a simple PLL, consisting of a phase detector, a low-pass filter (LPF), and a VCO. The PD serves as an "error amplifier" in the feedback loop, thereby minimizing the phase difference, $\Delta\phi$, between $x(t)$ and $y(t)$. The loop is considered "locked" if $\Delta\phi$ is constant with time, a result of which is that the input and output frequencies are equal.

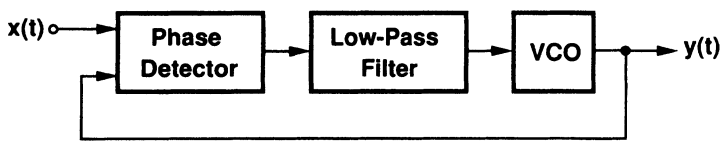

Fig. 11 Basic phase-locked loop.

In the locked condition, all the signals in the loop have reached a steady state and the PLL operates as follows. The phase detector produces an output whose dc value is proportional to $\Delta\phi$. The low-pass filter suppresses high-frequency components in the PD output, allowing the dc value to control the VCO frequency. The VCO then oscillates at a frequency equal to the input frequency and with a phase difference equal to $\Delta\phi$. Thus, the LPF generates the proper control voltage for the VCO.

It is instructive to examine the signals at various points in a PLL. Figure 12 shows a typical example. The input and output have equal frequencies but a finite phase difference, and the PD generates pulses whose widths are equal to the time difference between zero crossings of the input and output. These pulses are low-pass filtered to produce the dc voltage that sustains the VCO oscillation at the required frequency. As mentioned in Section 3.2, this voltage does not by itself determine the output *phase*. The VCO phase can be regarded as an initial condition of the system, independent of the initial conditions in the LPF.

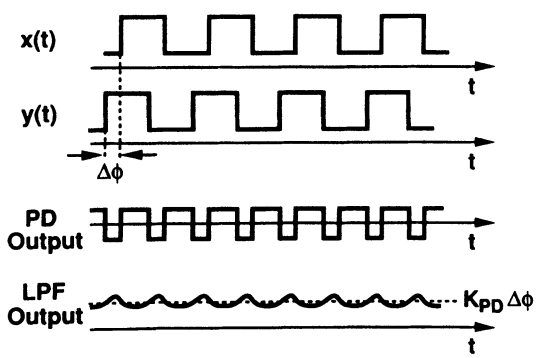

Fig. 12 Waveforms in a PLL.

Let us now study, qualitatively, the response of a PLL that is locked for $t < t_0$ and experiences a small, positive frequency step at the input at $t = t_0$ (Fig. 13). (For illustration purposes, the frequency step in this figure is only a few

percent.) We note that because the input frequency, ω_{in}, is momentarily greater than the output frequency, ω_{out}, $x(t)$ accumulates phase faster than does $y(t)$ and the PD generates increasingly wider pulses. Each of these pulses creates an increasingly higher dc voltage at the output of the LPF, thereby increasing the VCO frequency. As the difference between ω_{in} and ω_{out} diminishes, the width of the phase comparison pulses decreases, eventually returning to slightly greater than its value before $t = t_0$.

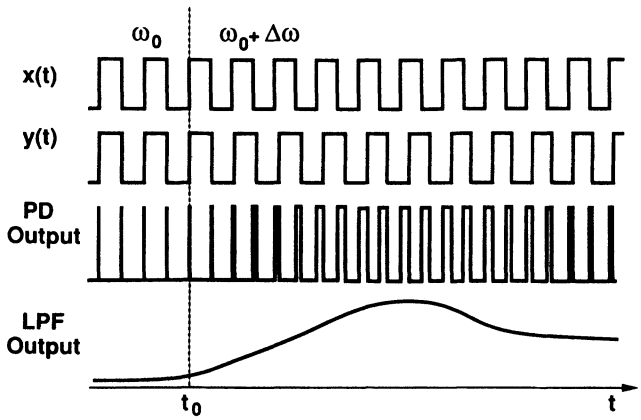

Fig. 13 Response of a PLL to a small frequency step.

The above analysis provides insight into the "tracking" capabilities of a PLL. If the input frequency changes slowly, its variation can be viewed as a succession of small narrow steps, during each of which the PLL behaves as in Figure 13.

It is important to note that in the above example the loop locks only after two conditions are satisfied: 1) ω_{out} has become equal to ω_{in}, and 2) the difference between ϕ_{in} and ϕ_{out} has settled to its proper value [1]. If the two frequencies become equal at a point in time but $\Delta\phi$ does not establish the required control voltage for the VCO, the loop must continue the transient, temporarily making the frequencies unequal again. In other words, both "frequency acquisition" and "phase acquisition" must be completed. This is, of course, to be expected because for lock to occur again, all the initial conditions of the system, including the VCO output phase, must be updated.

If the input to a PLL has a constant excess phase, i.e., is strictly periodic, but the input/output phase error, $\Delta\phi$, varies with time, we say the loop is "unlocked," an undesirable state because the output does not track the input or the relationship between the input and output is too complex to be useful. For example, if ω_{in} is sufficiently far from the VCO's free-running frequency, the loop may never lock. While the behavior of a PLL in the unlocked state is not important per se, whether and how it enters the locked state are both critical issues. Acquisition of lock is explained in Section 4.4.

Before studying PLLs in more detail, we make three important observations. First, because a PLL is a system with "memory," its output requires a finite time to respond to a change at its input, mandating a good understanding of the loop dynamics. Second, in a PLL, unlike many other feedback systems, the variable of interest changes dimension around the loop: it is converted from phase to voltage (or current) by the phase detector, processed by the LPF as such, and converted back to phase by the VCO. Third, in the lock condition, the input and output frequencies are *exactly* equal, regardless of the magnitude of the loop gain (although the phase error may not be zero). This is an extremely important property because many applications are intolerant of even small (systematic) differences between the input and output frequencies. Note that if the phase detector is replaced with only a frequency detector, this property vanishes.

While a PLL operates on phase, in many cases the parameter of interest is frequency. For example, we often need to know the response of the loop if 1) the input frequency is varied slowly, 2) the input frequency is varied rapidly, or 3) the input and output frequencies are not equal when the PLL is turned on. Therefore, the phase detector characteristic for unequal input frequencies plays an important role in the behavior of a phase-locked loop.

4.2 Loop Dynamics in Locked State

Transient response of phase-locked loops is generally a nonlinear process that cannot be formulated easily. Nevertheless, as with other feedback systems, a linear approximation can be used to gain intuition and understand trade-offs in PLL design.

Figure 14 shows a linear model of the PLL in lock along with the transfer function of each block. The model is to provide the overall transfer function for the phase, $\Phi_{out}(s)/\Phi_{in}(s)$; hence, the PD is represented by a subtractor. The LPF is assumed to have a voltage transfer function $G_{LPF}(s)$. The open-loop transfer function of the PLL is therefore equal to

$$H_O(s) = K_{PD}G_{LPF}(s)\frac{K_{VCO}}{s} \tag{17}$$

yielding the following closed-loop transfer function:

$$H(s) = \frac{\Phi_{out}(s)}{\Phi_{in}(s)} \tag{18}$$

$$= \frac{K_{PD}K_{VCO}G_{LPF}(s)}{s + K_{PD}K_{VCO}G_{LPF}(s)} \tag{19}$$

In its simplest form, a low-pass filter is implemented as in Figure 15, with

$$G_{LPF}(s) = \frac{1}{1 + \frac{s}{\omega_{LPF}}} \tag{20}$$

where $\omega_{LPF} = 1/(RC)$. Equation (19) then reduces to:

$$H(s) = \frac{K_{PD}K_{VCO}}{\frac{s^2}{\omega_{LPF}} + s + K_{PD}K_{VCO}} \tag{21}$$

indicating that the system is of second order, with one pole contributed by the VCO and another by the LPF. The quantity $K = K_{PD}K_{VCO}$ is called the *loop gain* and expressed in rad/s.

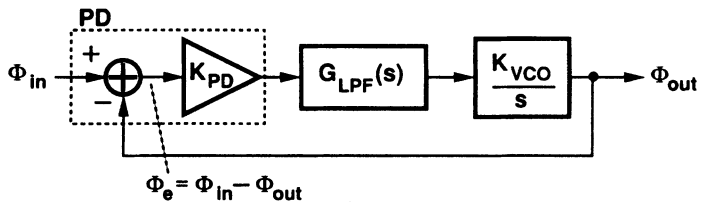

Fig. 14 Linear model of a PLL.

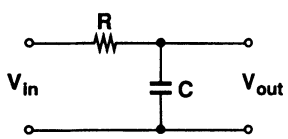

Fig. 15 Simple low-pass filter.

In order to understand the dynamic behavior of the PLL, we convert the denominator of Eq. (21) to the familiar form used in control theory: $s^2 + 2\zeta\omega_n s + \omega_n^2$, where ζ is the damping factor and ω_n is the natural frequency of the system.[3] Thus,

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{22}$$

where

$$\omega_n = \sqrt{\omega_{LPF}K} \tag{23}$$

$$\zeta = \frac{1}{2}\sqrt{\frac{\omega_{LPF}}{K}} \tag{24}$$

Note that ω_n is the geometric mean of the −3-dB bandwidth of the LPF and the loop gain, in a sense an indication of the gain-bandwidth product of the loop. Also, the damping factor is inversely proportional to the loop gain, an important and often undesirable trade-off that will be discussed.

In a well-designed second-order system, ζ is usually greater than 0.5 and preferably equal to $\sqrt{2}/2$ so as to provide an optimally flat frequency response. Therefore, K and ω_{LPF} cannot be chosen independently; for example, if $\zeta = \sqrt{2}/2$, then $K = \omega_{LPF}/2$. As explained in Sections 4.8 and 7.1, sideband or noise suppression issues typically impose an upper bound on ω_{LPF} and hence K. These limitations translate to significant phase error between the input and the output as well as a narrow capture range (Section 4.4).

The transfer function in Eq. (22) is that of a low-pass filter, suggesting that if the input excess phase varies slowly, then the output excess phase follows, and conversely, if the input excess phase varies rapidly, the output excess phase variation

[3] In a simple PLL, ω_n has no relation with the input and output frequencies.

will be small. In particular, if $s \to 0$, we note that $H(s) \to 1$; i.e., a *static* phase shift at the input is transferred to the output unchanged. This is because for phase quantities, the presence of integration in the VCO makes the open-loop gain approach infinity as $s \to 0$. To this end, we can examine the "phase error transfer function," defined as $H_e(s) = \Phi_e(s)/\Phi_{in}(s)$ in Figure 14:

$$H_e(s) = 1 - H(s) \tag{25}$$

$$= \frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{26}$$

which drops to zero as $s \to 0$.

Since phase and frequency are related by a linear, time-invariant operation, the transfer functions in Eqs. (22) and (26) also apply to the input and output excess frequencies. For example, Eq. (22) indicates that if the input frequency varies rapidly, the instantaneous variation of the output frequency will be small.

It is instructive to repeat our previous analysis of the loop step response (Fig. 13) with the aid of Eq. (22). Suppose the input excess frequency is equal to $\Delta\omega u(t)$, where $u(t)$ is the unit step function. The output excess frequency then exhibits the typical step response of a second-order system, eventually settling to $\Delta\omega$ rad/s higher than its initial value (Fig. 16). The output excess *phase*, on the other hand, is given by

$$\Phi_{out}(s) = H(s)\Phi_{in}(s) \tag{27}$$

$$= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{\Delta\omega}{s^2} \tag{28}$$

which is the response of a second-order system to a *ramp* input. More importantly, the phase error is

$$\Phi_e(s) = H_e(s)\Phi_{in}(s) \tag{29}$$

$$= \frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{\Delta\omega}{s^2} \tag{30}$$

whose final value is given by

$$\phi_e(t = \infty) = \lim_{s\to 0} s\phi_e(s) \tag{31}$$

$$= \Delta\omega\frac{2\zeta}{\omega_n} \tag{32}$$

$$= \frac{\Delta\omega}{K} \tag{33}$$

Therefore, static changes in the input frequency are suppressed by a factor of K when they manifest themselves in the static phase error (Fig. 16). This is, of course, to be expected because for the VCO frequency to change by $\Delta\omega$, the control voltage must change by $\Delta\omega/K_{VCO}$ and the input to the PD by $\Delta\omega/(K_{VCO}K_{PD})$.

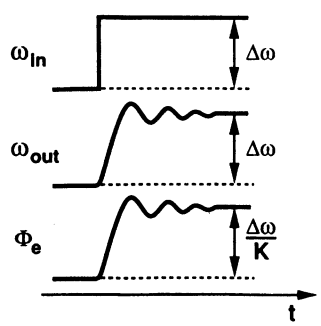

Fig. 16 Response of a PLL to a frequency step.

An important drawback of the PLL considered thus far is the direct relationship between ζ, ω_{LPF}, and K given by Eq. (24). For example, if the loop gain is increased to reduce the static phase error, then the settling behavior degrades. In order to allow independent choice of K and ω_{LPF}, a zero can be added to the low-pass filter as shown in Figure 17, modifying its transfer function to the following:

$$G_{LPF}(s) = \frac{R_2Cs + 1}{(R_1 + R_2)Cs + 1} \tag{34}$$

Thus, the PLL transfer function is

$$H(s) = \frac{K\left(\frac{s}{\omega_z}+1\right)}{\frac{s^2}{\omega_p}+\left(\frac{K}{\omega_z}+1\right)s+K} \tag{35}$$

$$= \frac{K\omega_p\left(\frac{s}{\omega_z}+1\right)}{s^2+\omega_p\left(\frac{K}{\omega_z}+1\right)s+K\omega_p} \tag{36}$$

where $\omega_z = 1/(R_2C)$ and $\omega_p = 1/[(R_1 + R_2)C]$. The damping factor is then equal to:

$$\zeta = \frac{1}{2}\sqrt{\frac{\omega_p}{K}}\left(\frac{K}{\omega_z}+1\right) \tag{37}$$

subject to the constraint $\omega_z > \omega_p$. As an example, if $\zeta = \sqrt{2}/2$, then with $\omega_z = \infty$, the open-loop gain is $K = \omega_p/2$, but with $\omega_z \approx 4.57\omega_p$, $K = 32\omega_p$.

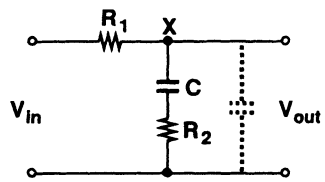

Fig. 17 LPF with a zero.

Note, however, that adding a zero to increase K has two side effects: 1) the −3-dB bandwidth of the system (roughly equal to $\omega_n = \sqrt{K\omega_p}$ for $\zeta = \sqrt{2}/2$) also increases, a trend that is desirable in some applications and undesirable in others; 2) the LPF attenuation of high-frequency signals is only $R_2/(R_1 + R_2)$, usually a troublesome drawback. To alleviate the latter effect, a second capacitor can be connected from node X in Figure 17 to ground so as to provide another pole beyond the zero (with some penalty in the settling time). (Because of the additional pole and zero, the optimum value of ζ may no longer be $\sqrt{2}/2$.)

In our discussion of PLLs thus far, we have assumed the feedback in the loop is negative. Interestingly, if the phase detector is realized with a multiplier, the polarity of feedback is unimportant. This is because the sinusoidal characteristic of the PD provides both negative and positive gains, allowing the loop to find a stable operating point by varying $\Delta\phi$. As illustrated in Figure 18, if $\Delta\phi$ begins around $+\pi/2$ and the feedback happens to be positive, the VCO frequency changes, driving $\Delta\phi$ toward $+3\pi/2$ or $-\pi/2$, where the PD gain has reverse polarity and the feedback is negative.

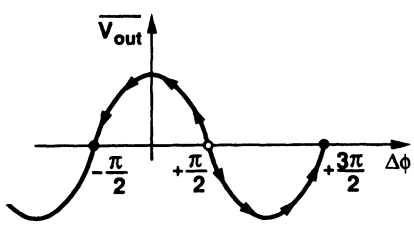

Fig. 18 Phase variation to provide negative feedback.

4.3 Tracking Behavior

The example of Figure 16 illustrates how a PLL can track the input frequency, indicating that in lock the input and output frequencies are equal but the phase error may not be zero. The natural question at this point is: How far can the PLL track the input frequency, i.e., what determines the "tracking range"[4] of the PLL? To answer this question, we consider two extreme cases: 1) the input frequency varies slowly (static tracking), and 2) the input frequency is changed abruptly (dynamic tracking). We will see that the tracking behavior is distinctly different in the two cases.

Suppose, starting from the VCO free-running frequency, the input frequency varies slowly such that the difference between ω_{in} and ω_{out} always remains much less than ω_{LPF} (or ω_p if the LPF contains a zero). Then, to allow tracking, the magnitude of the VCO control voltage, and hence the static phase error, must increase (Fig. 19). The PLL tracks as long as

[4]Also called the *lock range*.

the three parameters plotted in Figure 19 vary monotonically. In other words, the edge of the tracking range is reached at the point where the slope of one of the characteristics falls to zero or changes sign. This can occur only in the PD or the VCO (provided the LPF components are linear). Depicted in Figure 20 are examples of such behavior. The VCO frequency typically has a limited range, out of which its gain drops sharply. Also, in a typical phase detector, the characteristic becomes nonmonotonic for a sufficiently large input phase difference, at which point the PLL fails to maintain lock.

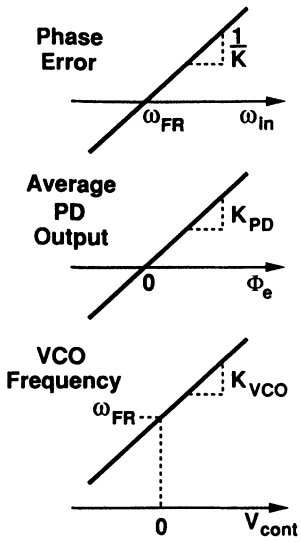

Fig. 19 Variation of parameters during tracking.

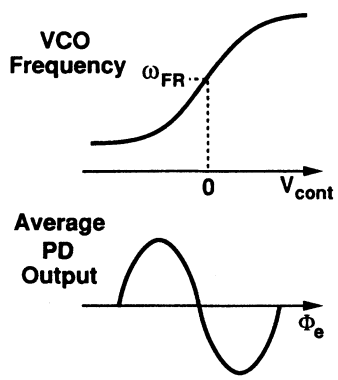

Fig. 20 Gain reduction in PD and VCO.

For a multiplier-type PD, as explained in Section 3.3, the gain, K_{PD}, changes sign if the input phase difference deviates from its center value by more than 90° (Fig. 10). Thus, the VCO output frequency can deviate from its free-running value by no more than

$$\Delta\omega_{tr} = K_{PD}\left(\sin\frac{\pi}{2}\right)K_{VCO} \tag{38}$$

Therefore, the static tracking range of a PLL employing a sinusoidal PD is the smaller of K and half of the VCO output frequency range.

Now let us study the tracking behavior of PLLs when the input frequency is changed abruptly. Suppose the input frequency of a PLL that is initially operating at $\omega_{in} = \omega_{out} = \omega_{FR}$ is stepped by $\Delta\omega$. What is the maximum $\Delta\omega$ for which the loop locks again? Can $\Delta\omega$ be as large as $\Delta\omega_{tr}$ in the case of static tracking [Eq. (38)]?

To answer these questions, we first make an important observation. Strictly speaking, we note that for *any* input frequency step at its input, a PLL loses lock, at least momentarily. This is evident from the simplified analysis in Figure 13, where the loop requires a number of cycles to stabilize. During these cycles, the input-output phase difference varies, and the PLL can be considered unlocked. (Nevertheless, for small $\Delta\omega$, the loop locks quickly, and the transient can be viewed as one of tracking rather than locking.)

The key point resulting from the above observation is that the following two situations are similar: 1) a loop initially locked at ω_{FR} experiences a large input frequency step, $\Delta\omega$; and 2) a loop initially unlocked and free running ($\omega_{out} = \omega_{FR}$) must lock onto an input frequency given by $|\omega_{in} - \omega_{FR}| = \Delta\omega$. In both cases, the loop must *acquire* lock.

4.4 Acquisition of Lock

The second case mentioned above occurs, for example, when a PLL is turned on. If the initial conditions in the LPF are zero, the VCO begins to oscillate at ω_{FR}, whereas the input is at a different frequency, $\omega_{FR} + \Delta\omega$. The "acquisition range" (also called the *capture range*) is the maximum value of $\Delta\omega$ for which the loop locks. To understand how a PLL acquires lock, we study the response from two different perspectives, namely, in the frequency and time domains. For simplicity, we make the following assumptions: a) the PD is implemented with a multiplier; b) ω_{in} is within the VCO frequency range; c) the sum component at the output of the mixer [the first term in Eq. (14)] is attenuated by the LPF to negligible levels; d) the VCO output frequency increases as its control voltage becomes more positive.

Consider the PLL shown in Figure 21, where initially $\omega_{in} = \omega_{FR} + \Delta\omega$ and $\omega_{out} = \omega_{FR}$. We begin with the top two spectra and follow the signals around the loop. At first glance, it may seem that, because $\omega_{in} \neq \omega_{out}$, the average output of the PD is zero and the loop cannot be driven toward lock. The important point, however, is that the LPF does not completely suppress the component at $\omega_{in} - \omega_{out} = \Delta\omega$. Thus, the VCO control voltage, V_A, varies at a rate equal to $\Delta\omega$, thereby modulating the output frequency:

$$v_{out}(t) = A\cos\left[\omega_{FR}t + K_{VCO}\int A_m\cos(\Delta\omega t)dt\right] \tag{39}$$

$$= A\cos\left[\omega_{FR}t + \frac{K_{VCO}}{\Delta\omega}A_m\sin(\Delta\omega t)\right] \tag{40}$$

$$\approx A\cos\omega_{FR}t - \frac{K_{VCO}}{\Delta\omega}A_m\sin\omega_{FR}t\sin(\Delta\omega t) \tag{41}$$

where we have assumed $K_{VCO}A_m/\Delta\omega \ll 1$. As a result, the VCO output, V_B, exhibits sidebands at $\omega_{FR} \pm \Delta\omega$ in addition to the main component at ω_{FR}. When the PD multiplies the sideband at $\omega_{FR} + \Delta\omega$ by ω_{in}, a dc component appears at node A (Fig. 21), adjusting the VCO frequency toward lock [1]. The dc component may need to grow over a number of beat cycles before lock is achieved.

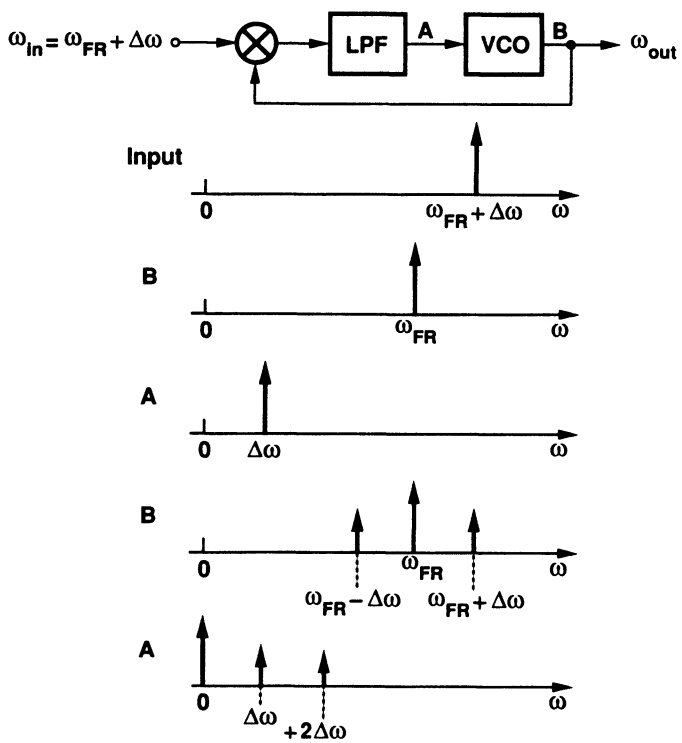

Fig. 21 Acquisition behavior in frequency domain.

From the above example, we note that the acquisition range depends on how much the LPF passes the component at $\Delta\omega$ and how strong the feedback dc component is, i.e., the acquisition range is a direct function of the loop gain at $\Delta\omega$. In other words, because the loop gain of a simple PLL drops as the difference between the input frequency and the VCO frequency increases, the acquisition range cannot be arbitrarily wide.

A second approach to analyzing the acquisition behavior is in the time domain. First, suppose the loop is opened at the VCO output and the feedback signal is replaced by a source oscillating at ω_{FR} (Fig. 22a) [2]. The output of the LPF is then a sinusoid at $\omega_{in} - \omega_{FR}$. As the sinusoid instantaneous amplitude increases, so does the VCO frequency and vice versa. Consequently, the difference between ω_{in} and ω_{out} reaches a maximum when the sinusoid is at a positive peak and a minimum when the sinusoid is at a negative peak. (This is simply the modulation phenomenon of Fig. 21 described in the time domain.) Now, if the loop is closed, the feedback signal has a *time-varying* frequency. When the LPF output goes through a positive excursion, ω_{out} approaches ω_{in} and the beat period increases. Conversely, when the LPF output becomes negative, ω_{out} moves away from ω_{in} and the beat period decreases. Shown in Fig. 22b, the resulting waveform at the LPF output exhibits longer positive cycles than negative ones, thus carrying a positive dc component and gradually shifting the average value of ω_{out} closer to ω_{in}.

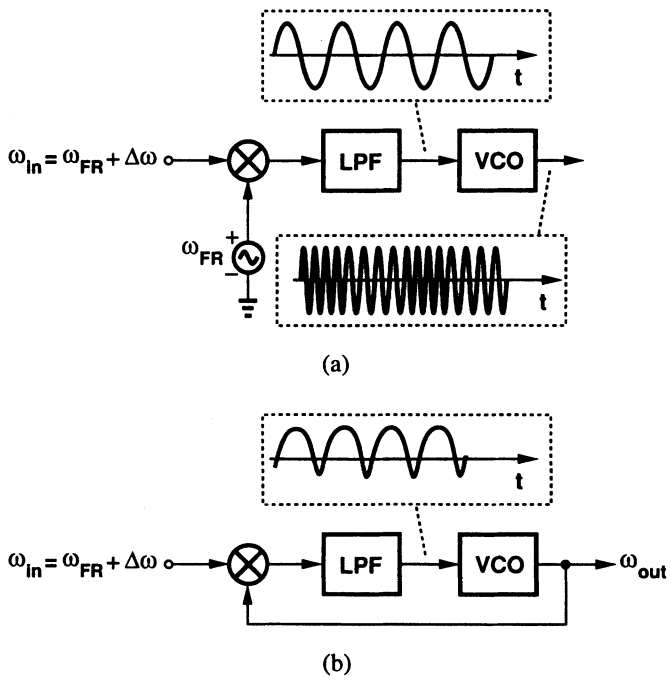

Fig. 22 Acquisition behavior in time domain.

The above time-domain analysis reveals two important points. First, if ω_{in} is sufficiently close to ω_{FR}, frequency acquisition is achieved at the first proper peak of the beat waveform (Fig. 23a). In this case, we say the PLL has locked with no "cycle slips." Second, if ω_{in} is sufficiently far from ω_{FR}, the beat waveform has little asymmetry and hence not enough dc voltage to drive the loop toward lock (Fig. 23b).

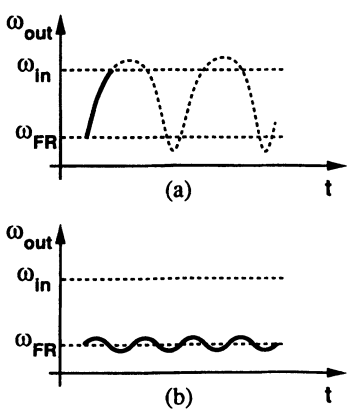

Fig. 23 (a) Capture with no cycle clips; (b) capture failure.

Figure 24 illustrates the simulated acquisition behavior of a 1-GHz PLL. Plotted here is the control voltage of the VCO as a function of time, exhibiting several cycle slips before the loop enters small-signal settling. While it is difficult to

see in this figure, the peak of the beat cycles gradually becomes more positive and the period of each cycle slightly increases as the average value of ω_{out} comes closer to ω_{in}. Note, however, that cycle slips are observed only if ω_{in} is very close to the edge of the acquisition range.[5] Also, the number of cycle slips depends on the loop's initial conditions, i.e., those in the LPF as well as the initial phase of the VCO.

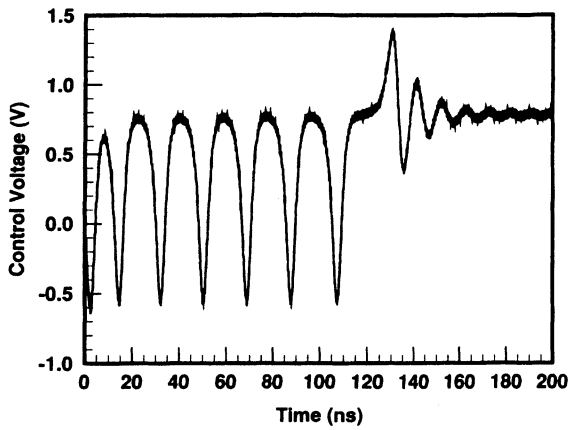

Fig. 24 Simulated capture behavior.

Acquisition range is a critical parameter because 1) it trades directly with the loop bandwidth; i.e., if an application requires a small loop bandwidth, the acquisition range will be proportionally small; 2) it determines the maximum frequency variation in the input or the VCO that can be accommodated. In monolithic implementations, the VCO free-running frequency can vary substantially with temperature and process, thereby requiring a wide acquisition range even if the input frequency is tightly controlled.

Unfortunately, it is difficult to calculate the acquisition range of PLLs analytically. To gain a better feeling about the limits, we consider a simplified case where the LPF output signal can be approximated as [2]

$$v_{LPF}(t) = K_{PD}|G_{LPF}(j\Delta\omega)|\sin(\Delta\omega t) \tag{42}$$

This signal modulates the VCO frequency, causing a maximum *deviation* of

$$(\omega_{out} - \omega_{FR})|_{max} = K_{PD}K_{VCO}|G_{LPF}(j\Delta\omega)| \tag{43}$$

As shown in Figure 23, if this deviation is equal to or greater than $\Delta\omega$, then the loop locks without cycle slips [3]:

$$\Delta\omega_{acq} = K_{PD}K_{VCO}|G_{LPF}(j\Delta\omega)| \tag{44}$$

With $G_{LPF}(s)$ known, $\Delta\omega$ can be calculated from this equation. For a simple low-pass filter:

$$\Delta\omega_{acq} = \left[\frac{\omega_{LPF}^2}{2}\left(-1+\sqrt{1+\frac{1}{4\zeta^4}}\right)\right]^{1/2} \tag{45}$$

which reduces to $\Delta\omega_{acq} \approx 0.46\omega_{LPF}$ if $\zeta = \sqrt{2}/2$.

The above derivation actually underestimates the capture range. This is because as ω_{in} is brought closer to ω_{FR}, the "average" frequency of the VCO also departs from ω_{FR} and comes closer to ω_{in} [4]. Thus, in the vicinity of lock, the difference between ω_{in} and ω_{VCO} is small and the LPF attenuation predicted by Eq. (42) too large. A more accurate expression is given in [5].

Most modern phase-locked systems incorporate additional means of frequency acquisition to significantly increase the capture range, often removing its dependence on K and ω_{LPF} and achieving limits equal to those of the VCO. This is discussed below.

4.5 Acquisition Time

The acquisition and settling times of PLLs are important in many applications. For example, if a PLL is used at the clock interface of a microprocessor (Fig. 2) and the system is powered down frequently to save energy, it becomes critical to know how long the system must remain idle after it is turned on to allow adequate phase alignment between the external and internal clocks. As another example, when a frequency synthesizer used in a wireless transceiver is switched to change its output frequency, the resulting loop transient causes "frequency spreading," in effect leaking noise into other channels. Furthermore, because channel spacing in such an environment is typically several orders of magnitude smaller than the VCO frequency range, precise settling of the loop (e.g., a few parts per million) is required.

[5]Or if the input is heavily corrupted by noise [1].

For a simple second-order system with $\zeta < 1$, the step response is expressed as

$$y(t) = \left[1 + \frac{1}{\sqrt{1-\zeta^2}} \exp(-\zeta\omega_n t) \times \sin\left(\omega_n\sqrt{1-\zeta^2}t - \psi\right)\right] u(t) \tag{46}$$

where $\psi = \sin^{-1}\sqrt{1-\zeta^2}$. Thus, the decay time constant is

$$\tau_{dec} = \frac{1}{\zeta\omega_n} \tag{47}$$

$$= \frac{2}{\omega_{LPF}} \tag{48}$$

and the frequency of ringing equals $\omega_n\sqrt{1-\zeta^2}$. For a frequency step at the PLL input, Eq. (46) can be used to calculate the time required for the output frequency to settle within a given error band around its final value.

Note that Eq. (46) assumes a linear system. In practice, nonlinearities in K_{PD} and K_{VCO} result in somewhat different settling characteristics, and simulations must be used to predict the lock time accurately. Nonetheless, this equation provides an initial guess that proves useful in early phases of the design.

4.6 Aided Acquisition

The acquisition behavior in Figure 23 and formulated by Eq. (45) indicates that the capture range of a simple, optimally stable phase-locked loop is roughly equal to $0.5\omega_{LPF}$, regardless of the magnitude of K. Since issues such as jitter and sideband suppression impose an upper bound on ω_{LPF}, the resulting capture range is often inadequate. Therefore, most practical PLLs employ additional techniques to aid the acquisition of frequency.

Shown in Figure 25 is a conceptual diagram of a PLL with aided frequency acquisition. Here, the system utilizes a frequency detector (FD) and a second low-pass filter, LPF_2, whose output is added to that of LPF_1. The FD produces an output having a dc value proportional to and with the same polarity as $\omega_{in} - \omega_{out}$. If the difference between ω_{in} and ω_{out} is large, the PD output has a negligible dc component and the VCO is driven by the dc output of the FD with negative feedback, thereby moving ω_{out} toward ω_{in}. As $|\omega_{in} - \omega_{out}|$ drops, the dc output of the FD decreases, whereas that of the PD increases. Thus, the frequency detection loop gradually relinquishes the acquisition to the phase-locked loop, becoming inactive when $\omega_{in} - \omega_{out} = 0$.

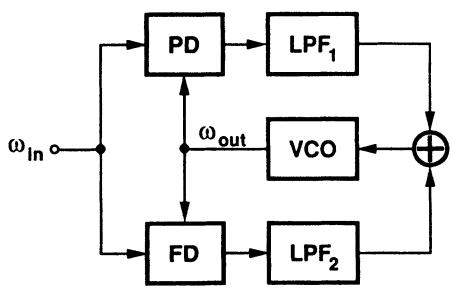

Fig. 25 Aided acquisition with a frequency detector.

It is important to note that in a frequency detection loop, the loop gain is relatively constant, independent of $|\omega_{in} - \omega_{out}|$, whereas in a simple phase-locked loop it drops if $|\omega_{in} - \omega_{out}|$ exceeds ω_{LPF}. For this reason, aided acquisition using FDs can substantially increase the capture range.

The configuration of Figure 25 is greatly simplified if a single circuit can perform both frequency and phase detection. This is discussed in Section 5.2.

4.7 Higher Order Loops

The generic phase-locked loop considered thus far is of second order. In principle, the low-pass filter can include more poles to achieve sharper cut-off characteristics, a desirable property in many applications. However, such systems are difficult to stabilize, especially when process and temperature variations are taken into account. On the other hand, in many cases the PLL inevitably has a third pole, for example, if a capacitor is connected in parallel with the LPF output port (Fig. 17) to suppress high frequencies. Thus, most practical PLLs can be considered as third-order topologies with the third pole being much farther from the origin than the other two.

4.8 Frequency Multiplication

Phase-locked loops are often used in applications where the output frequency must be a multiple of the input frequency. The design problem illustrated in Figure 2, for example, sets an upper bound on the external clock frequency because of signal distribution issues on PC boards, but it also requires higher internal clock frequencies for the processor.

A PLL can "amplify" a frequency in the same fashion as does a feedback amplifier. As shown in Figure 26a, to amplify the input, the output signal is divided down before it is fed back. Since the output quantity of interest in a PLL is the frequency, a frequency divider (e.g., a digital counter) must be inserted in the feedback loop (Fig. 26b). From another perspective, when the loop is locked, $\omega_F = \omega_{in}$, and hence $\omega_{out} = M\omega_{in}$.[6]

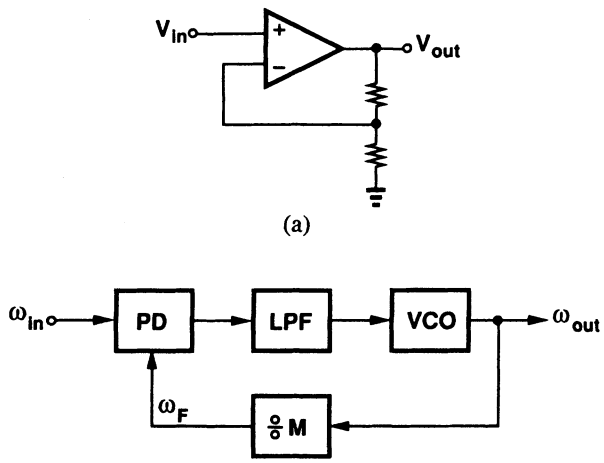

Fig. 26 Signal "amplification" in (a) a feedback amplifier; (b) a PLL.

The analogy depicted in Figure 26 also proves useful in studying the effect of the $\div M$ circuit upon the PLL behavior. As with the feedback amplifier, the loop gain is divided by M and hence the results of all of the previous static and dynamic analyses can be directly applied if K is replaced by K/M. Before examining the consequences of this change, we need to make an important observation.

Consider the system shown in Figure 26b with the phase detector implemented as a multiplier. When the loop is locked, the PD output consists of two components: one at $\omega_{in} - \omega_{out}/M = 0$ and another at $\omega_{in} + \omega_{out}/M = 2\omega_{in}$. Since the LPF has a finite stopband attenuation, the VCO control voltage contains a frequency component at $2\omega_{in}$, thereby modulating the output frequency and creating sidebands at $M\omega_{in} \pm 2\omega_{in}$. As mentioned in Section 3.1, the effect of these sidebands in the time domain can be viewed as jitter for most practical purposes. Furthermore, in applications such as wireless transceivers, the sidebands must be several orders of magnitude smaller than the carrier. Thus, ω_{LPF} must be chosen so as to sufficiently attenuate the component at $2\omega_{in}$. In other words, for a given ω_{out}, higher division ratios translate into lower ω_{LPF}. For simplicity, we assume the LPF cut-off frequency scales inversely with M.

With a divider in the loop and a fixed ω_{out}, the damping factor is:

$$\zeta = \frac{1}{2}\sqrt{\frac{\omega_{LPF}/M}{K/M}} \tag{49}$$

$$= \frac{1}{2}\sqrt{\frac{\omega_{LPF}}{K}} \tag{50}$$

and the natural frequency:

$$\omega_n = \sqrt{\frac{K}{M}\frac{\omega_{LPF}}{M}} \tag{51}$$

$$= \frac{1}{M}\sqrt{K\omega_{LPF}} \tag{52}$$

It follows from Eq. (52) that the settling is slowed down by a factor M.

It is interesting to note that frequency multiplication in Figure 26 also amplifies the input jitter. For example, jitter frequencies below the −3-dB bandwidth of the PLL are amplified by a factor M.

4.9 Delay-Locked Loops

A close relative of phase-locked loops is the delay-locked loop (DLL) [6, 7]. Shown in Figure 27a, a DLL replaces the VCO of a PLL with a voltage-controlled delay line (VCDL). The idea is that if a periodic input is delayed by an integer

[6]While the feedback amplifier suffers from gain error (due to the finite open-loop gain of the op amp), the PLL exhibits no such behavior. We leave the explanation as an exercise for the reader.

multiple of the period, T_{in}, then its phase shift can be considered zero. Thus, the phase detector drives the loop so that the phase difference between V_{in} and V_{out} approaches nT_{in}, where n is an integer (in most cases equal to unity). Note that the polarity of feedback must be negative.

The VCDL usually consists of a cascade of k identical gain stages with variable delay, as shown in simple form in Figure 27b. (Most of the VCO design issues described in Section 5.1 apply to VCDLs as well.) Note that, unlike oscillators, delay lines do not *generate* a signal, making it difficult to perform frequency multiplication in a DLL.

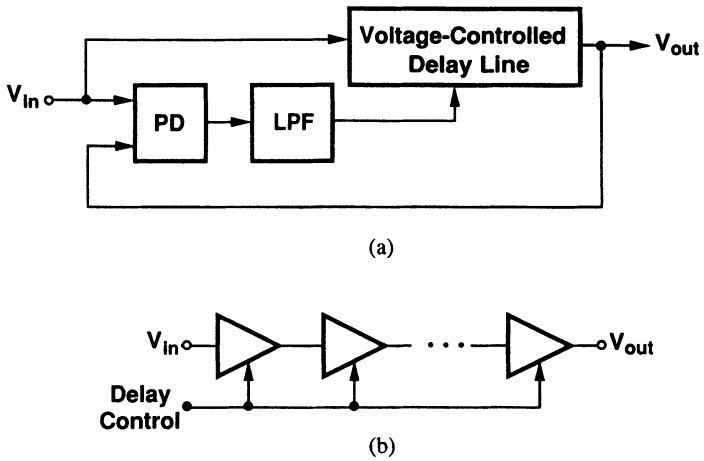

Fig. 27 (a) DLL block diagram; (b) delay line.

In addition to phase alignment, DLLs can provide precisely spaced timing edges even in the presence of temperature and process variations. To understand this, note that in Figure 27b, the delay between V_{in} and V_{out} is equal to nT_{in} when lock is achieved. Therefore, output signals of consecutive stages exhibit a phase difference of nT_{in}/k, a value independent of device parameters. In practice, mismatches between stages limit the edge delay accuracy.

DLLs have two important advantages over PLLs. First, because the delay line has no "memory," its transfer function is a constant, thereby yielding a *first-order* open-loop transfer function for the entire system (for a first-order LPF). Consequently, DLLs have much more relaxed trade-offs among gain, bandwidth, and stability. Second, delay lines typically introduce much less jitter than oscillators. Intuitively, this is because delaying a signal entails much less uncertainty than *generating* it. From another point of view, noise injected into a DLL disappears at the end of the delay line, whereas it is recirculated in an oscillator [8].

As DLLs can lock with a total delay of nT_{in}, some means must be provided to obtain the desired n. For $n = 1$, the maximum delay of the VCDL must remain less than $2T_{in}$ [7].

5. Building Blocks of PLLs

While the generic PLL architectures of Figures 11 and 25 have been extensively used with little modification, their building blocks have been implemented in many different forms. Innovations in the design of these building blocks have tremendously improved the speed, power dissipation, jitter, and capture range of phase-locked systems. In this section, we describe the design of voltage-controlled oscillators, phase and frequency detectors, and charge pumps.

5.1 Voltage-Controlled Oscillators

Perhaps the most critical part of PLLs and CRCs, oscillators have been the subject of numerous studies for more than half a century [9, 10], still defying an exact analysis. Most of these analyses have considered only "nearly sinusoidal" oscillations in conventional topologies (such as LC-based circuits), conditions difficult to create in monolithic circuits. For this reason, before our knowledge in this area advances sufficiently, we must rely on simulations to predict various parameters of oscillators (as far as simulations can go).

For a VCO that is to be used in a PLL, the following parameters are important. 1) Tuning range: i.e., the range between the minimum and maximum values of the VCO frequency. In this range, the *variation* of the output amplitude and jitter must be minimal. The tuning range must accommodate the PLL input frequency range as well as process- and temperature-induced variations in the VCO frequency range. The tuning range is typically at least $\pm 20\% \omega_{FR}$. 2) Jitter and phase noise: timing accuracy and spectral purity requirements in PLL applications impose an upper bound on the VCO jitter and phase noise. 3) Supply and substrate noise rejection: if integrated along with digital circuits, VCOs must be highly immune to supply and substrate noise. In the architecture of Fig. 26b, for example, the frequency divider can corrupt the

VCO output by injecting noise into the common substrate. Such effects become more prominent if a PLL shares the same substrate and package with a large digital processor. 4) Input/output characteristic linearity: variation of K_{VCO} across the tuning range is generally undesirable. If a PLL is used as an FM demodulator, the variation of K_{VCO} introduces harmonic distortion in the detected signal and must be below 1%. In other applications, this nonlinearity degrades the loop stability but it can be as high as several tens of percent.

Before describing VCO topologies, we must explain an extremely important point: in order to achieve high rejection of supply and substrate noise, both the signal path *and* the control path of a VCO must be fully differential. We also note that an oscillator in which signals exist in complementary form but have rail-to-rail swings is not considered differential because it exhibits poor supply rejection. Differential operation also yields a 50% duty cycle, an important requirement in timing applications, and is immune to the up-conversion of low-frequency noise components in the signal path [11].

A common oscillator topology in monolithic PLLs is the ring oscillator, shown in Figure 28. Here, a cascade of M gain stages with a total (dc) phase shift of 180° is placed in a feedback loop. It can be easily shown that the loop oscillates with a period equal to $2MT_d$, where T_d is the delay of each stage with a fanout of one. The oscillation can also be viewed as occurring at the frequency for which the total phase shift is zero and the loop gain is unity. Since in a typical IC technology the gate delay is monitored and controlled within the process corners, the oscillator frequency and its variation can be predicted with reasonable accuracy.

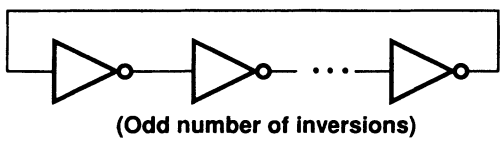

Fig. 28 Ring oscillator.

The gain stages in a ring oscillator can be implemented in various forms, some of which are shown in Figure 29. Note that with the differential pairs of Figures 29b and c, the number of stages in the ring need not be odd; the total phase shift can be changed by 180° if the output signals of one of the stages are swapped. With an even number of stages, the oscillator can provide quadrature outputs, i.e., outputs that are 90° out of phase.

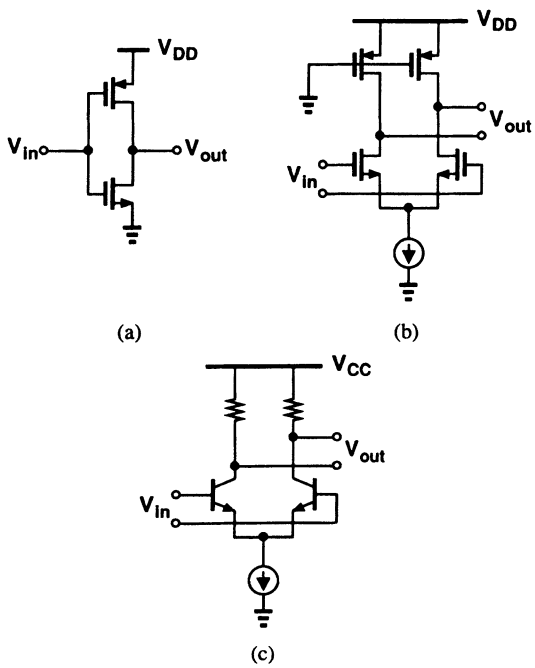

Fig. 29 Simple gain stages.

In order to vary the frequency of oscillation, one of the parameters in $2MT_d$ must change, i.e., the effective number of stages or the delay of each stage. The resulting techniques are conceptually illustrated in Figure 30. In the first approach, called "delay interpolation," a fast path and a slow path are used in parallel [12, 13]. The total delay is adjusted by increasing the gain of one path and decreasing that of the other, and hence is a weighted sum of the delays of the two paths. In the second approach, the delay of each stage in the ring is directly varied with negligible change in the gain or voltage swings.

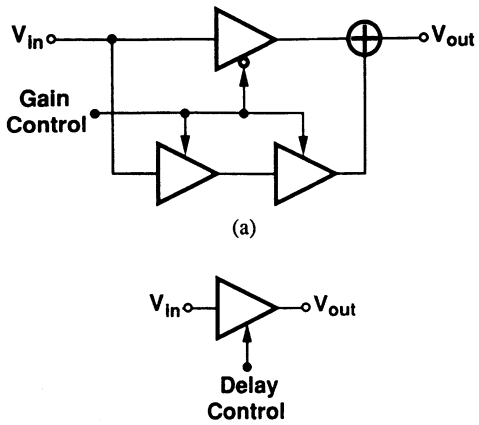

Fig. 30 Frequency variation techniques.

A simple implementation of delay interpolation is shown in Figure 31. The pair Q_1-Q_2 constitutes the fast path and the pairs Q_3-Q_4 and Q_5-Q_6, the slow path. The control input, V_{cont}, determines the gain of each path by steering I_{EE} from one to the other. The same topology can be used with CMOS devices.

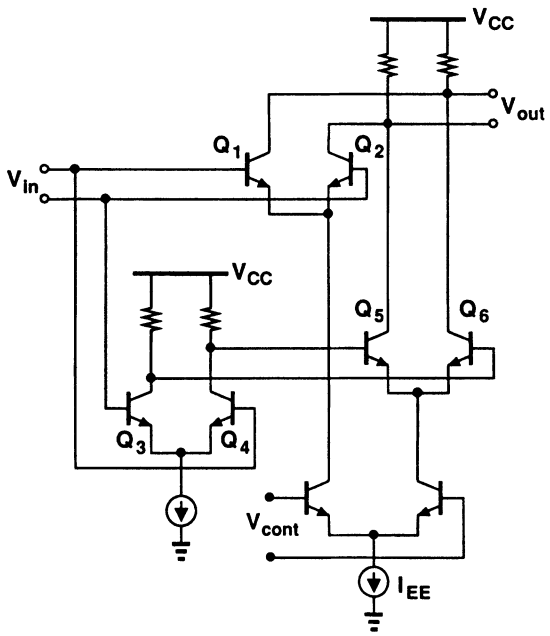

Fig. 31 Delay interpolation in a bipolar VCO.

In Figure 30b, the delay of each stage is tuned by the control input. This can be accomplished by varying the capacitance or the resistance seen at the output node of each stage. We first consider "capacitive tuning" to explain its drawbacks and lead to "resistive tuning" as the superior technique.

In order to vary the effective capacitance seen at a node, one of the two methods depicted in Figure 32 can be used. In Figure 32a, a voltage-dependent capacitor, for example, a reverse-biased pn junction diode, loads node X, and its value is adjusted by V_{cont}. The drawback is that the minimum value of C_1 (usually determined by the maximum range of V_{cont}) still loads the circuit, limiting the maximum frequency of operation. In Figure 32b, C_1 is constant but a MOS device, M_1, operates as a voltage-dependent resistor, thereby varying the "effective" capacitance seen at node X. The difficulty here is that K_{VCO} can experience substantial variation, especially if a wide tuning range is required. Additionally, both techniques use a single-ended control and are therefore susceptible to common-mode noise.

In contrast to capacitive tuning, resistive tuning provides a large, relatively uniform frequency variation and lends itself to differential control. Nonetheless, in addition to the effective load resistance, some other parameters of each stage must also be varied so that the voltage swings and/or the (dc) voltage gain remain relatively constant. To understand this issue, consider the gain stages shown in Figure 33. In Figure 33a, the load devices are biased in the triode region and their on-resistance is adjusted by V_{cont}. As V_{cont} decreases, the delay of the stage drops because the time constant at the output

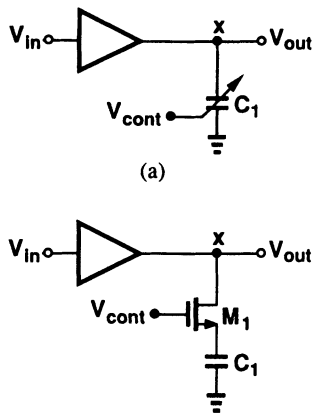

Fig. 32 Capacitive tuning.

nodes decreases. Note that the small-signal gain also decreases. The variable voltage swings present a non-optimal solution: when they are small, the signals are more corrupted by jitter, and when they are large, they require a higher supply voltage so as to preserve differential operation. Moreover, as the gain of each stage drops, the circuit eventually fails to oscillate because the total gain around the ring *at the frequency of oscillation* falls below unity. This is particularly problematic in CMOS oscillators if they employ a small number of stages and a low gain in each stage to achieve a high speed.

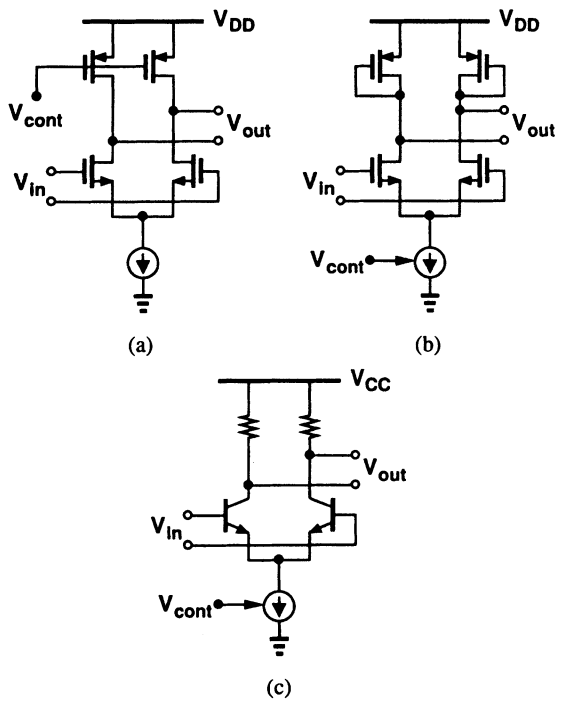

Fig. 33 Resistive tuning.

In Figure 33b, on the other hand, as V_{cont} adjusts the tail current, the small-signal impedance of the load devices varies accordingly, but the voltage gain remains relatively constant. Thus, the circuit may seem to be appropriate for a VCO stage. However, the large-signal output voltage swings still depend on the current because $I_D = \mu_p C_{ox} W(V_{GS} - V_{THP})^2/(2L)$.

It is also important to understand why a configuration such as that in Figure 33c is a poor choice for a variable-delay stage. Since the time constant at the collector of the bipolar transistors does not directly depend on the tail current, the tuning range of an oscillator employing this stage is very small. (In reality, the input capacitance of each stage increases with the tail current, but the time for which each stage is on decreases. Thus, the "average" input capacitance remains relatively constant.)

Another approach to varying the delay of the stages in a ring oscillator is to vary the effective resistance using local positive feedback. Feasible in both bipolar and CMOS technologies, this method is illustrated in a CMOS circuit in Figure 34. Here, the cross-coupled pair M_5-M_6 introduces a *negative* average resistance equal to $-2/g_m$, where g_m denotes the average transconductance of M_5 and M_6. This resistance partially cancels that seen at the drain of M_3 and M_4, increasing the effective

output impedance and hence the delay. The key point is that the total current flowing from M_3 and M_4, and thus the voltage swings at X and Y remain constant as V_{cont} steers I_{SS} between M_1-M_2 and M_5-M_6.

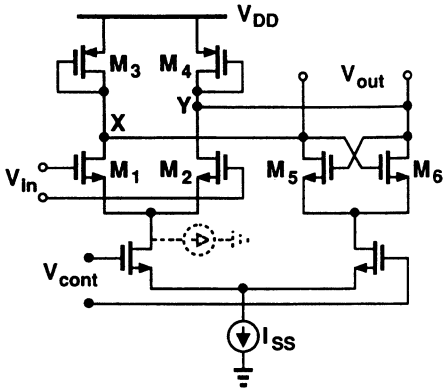

Fig. 34 Delay variation using local positive feedback.

In designing the circuit of Figure 34, two issues must be borne in mind. First, to avoid latch-up, the transconductance of M_5-M_6 must be less than that of M_3-M_4, a condition met if the latter devices are wider than the former and all four have equal lengths. Second, to ensure steady oscillations, the input pair, M_1-M_2, must have an additional constant bias current so that the stage has adequate gain even if a PLL transient steers all of I_{SS} to M_5-M_6.

While providing at least a two-to-one frequency range, this topology cannot operate from a 3-V supply if implemented in standard CMOS technology. This is due to the large gate-source voltages (including the body effect) of the stacked transistors. Figure 35 shows how two modifications can reduce the minimum supply voltage to approximately 2.5 V. First, the diode-connected PMOS loads are replaced with a composite PMOS/NMOS structure that exhibits approximately the same impedance but consumes less voltage headroom owing to the level shift provided by the NMOS source follower. Second, the control path is implemented as a folded PMOS stage to avoid stacking.

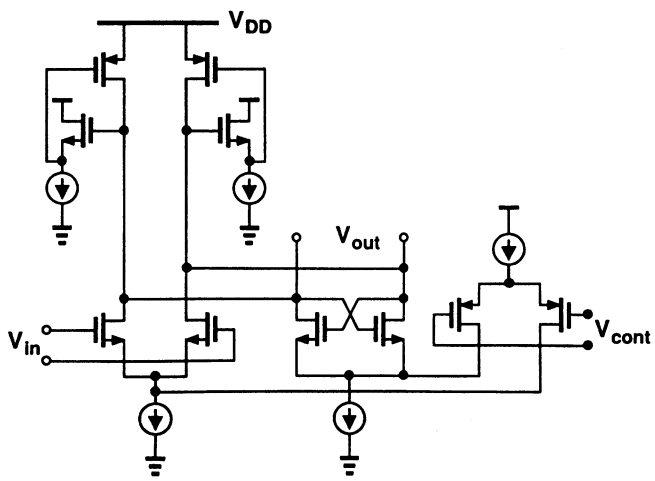

Fig. 35 Low-voltage variable-delay gain stage.

In bipolar technology, in addition to delay interpolation and local positive feedback, other techniques can be used to achieve a wide tuning range [14, 15].

An important issue in ring oscillator design is the minimum number of stages that can be used while attaining reliable operation. Since oscillation occurs at a frequency for which the total phase shift is zero and the loop gain is unity, as the number of stages decreases, the required phase shift and (dc) gain per stage increase. For example, for a three-stage oscillator, each stage must introduce a phase shift of 120° and a minimum dc gain of 2 [11]. While two-stage bipolar oscillators can be designed to achieve both sufficient phase shift and high speed [14, 15], simulations show that CMOS implementations with only two stages either do not operate reliably or, if they employ additional phase shift elements, oscillate no faster than three-stage configurations. Thus, CMOS VCOs typically utilize three or more stages.

5.2 Phase and Frequency Detectors

The PLL analysis in Section 4 indicates that many parameters of phase-locked systems, including tracking range, acquisition range, loop gain, and transient response depend on the properties of the phase and frequency detectors employed

in such systems. Of particular interest are the following properties: 1) what is the input-phase difference range for which the characteristic is monotonic? 2) What is the response to unequal input frequencies? 3) How do the input amplitude and duty cycle affect the characteristic?

Gilbert Cell. A "combinational" phase detector often used in PLLs is the Gilbert cell (Fig. 36). For small signals applied to ports A and B, the circuit operates as an analog multiplier, with an average output given by Eq. (15). Note that as $\Delta\phi$ departs from 90°, the slope of cos $\Delta\phi$ and, hence, the equivalent K_{PD} decrease. For $\omega_1 \neq \omega_2$, the average output is zero; i.e., the circuit cannot be used as a frequency detector. Also, K_{PD} is a function of A and B, an undesirable attribute because a PLL employing such a phase detector exhibits amplitude-dependent static and dynamic behavior.

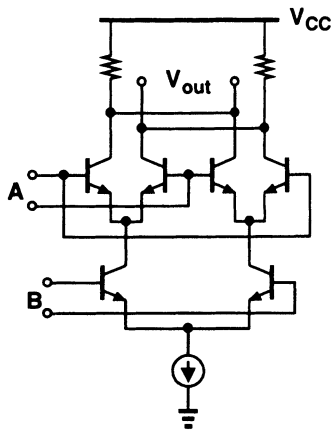

Fig. 36 Gilbert cell PD.

In the Gilbert cell of Figure 36, if the input signal amplitudes are much greater than kT/q, then all three differential pairs experience full switching and the circuit operates as an exclusive OR (XOR) gate. In this case, we consider the input waveforms to be triangular and examine the average output as the input phase difference varies (Fig. 37). If the two inputs are 90° out of phase (Fig. 37a), the output is a square wave with zero dc. As the phase difference deviates from 90°, the output duty cycle is no longer 50%, providing a dc value proportional to the phase difference. Thus, in contrast to the small-signal multiplier, an XOR gate has a constant gain for $0 < |\Delta\phi| < 180°$ (Fig. 38). It can be easily shown that this translates to a factor of $\pi/2$ increase in the tracking range of PLLs. For this reason, and because K_{PD} is independent of the signal amplitudes, it is preferable to use a Gilbert cell with large signals.

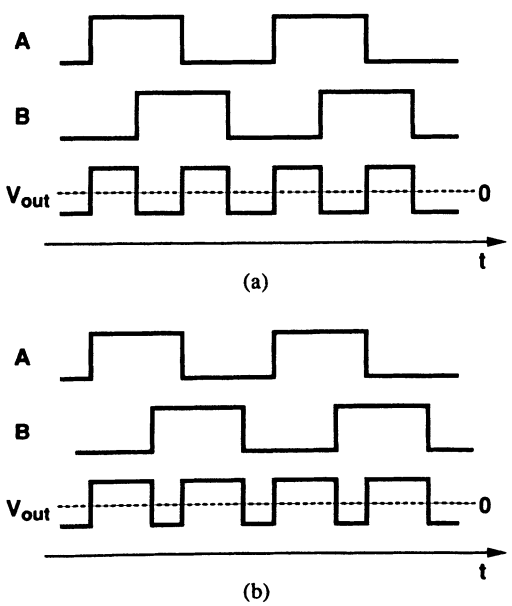

Fig. 37 Input and output waveforms of an XOR gate.

It is interesting to note that in a PLL with a large loop gain, a Gilbert-type PD forces the static phase difference between the input and output to remain close to 90° so that the dc output of the PD is small. In many applications, the 90° phase shift is unimportant or can be cancelled elsewhere in the system.

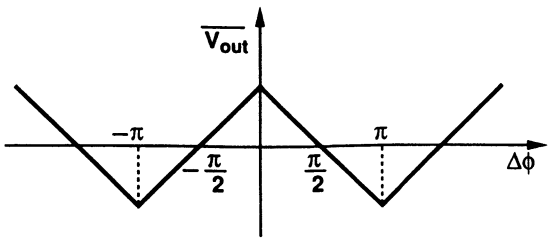

Fig. 38 Characteristic of an XOR PD.

In the Gilbert cell PD, the average output depends on the duty cycle of the inputs. Illustrated in Figure 39, this effect manifests itself as a static phase error in a PLL.

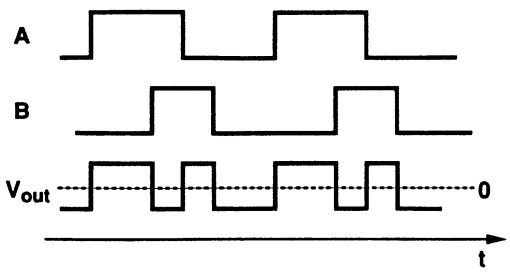

Fig. 39 Dependence of XOR output on input duty cycle.

R-S Latch. Another topology that can be used for phase detection is an edge-triggered R-S latch, also called a two-state PD (Fig. 40). Here, the rising edge of A drives Q to *ONE* and that of B drives Q to *ZERO*. Thus, the differential output changes sign every time a rising edge at one input is followed by a rising edge at the other (Fig. 40b). Since this circuit changes state only on one edge of the inputs, its characteristics differ from those of an XOR in several respects: 1) the output frequency is the same as the input frequency; 2) the average output does not depend on the input duty cycle; 3) the input/output characteristic crosses zero when the inputs are 180° out of phase (Fig. 40b); 4) the monotonic range of the PD is ±180° around the center; 5) the shape of the characteristic is sawtooth rather than triangular. Among these, the first property is undesirable if the PLL performs frequency multiplication because it mandates a low −3-dB frequency in the LPF (Section 4.8).

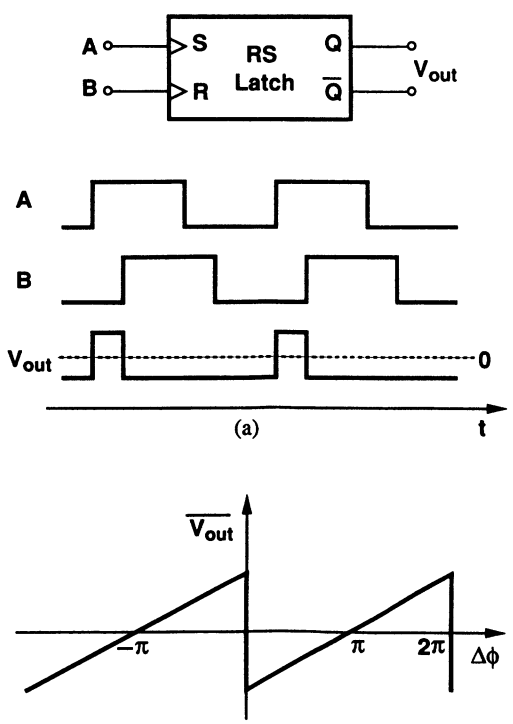

Fig. 40 Edge-triggered R-S latch as a PD.

It is also interesting to note that an R-S latch generates a nonzero dc output if one input frequency is an integer multiple of the other (Fig. 41), whereas an XOR gate exhibits no such behavior. Thus, a PLL employing an R-S latch as the PD may lock to a higher harmonic of the input if the VCO frequency range is sufficiently wide or the input spectrum has strong components at subharmonics of the VCO output.

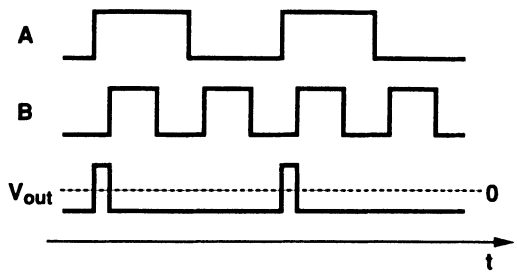

Fig. 41 Response of R-S latch to higher harmonics.

Phase/Frequency Detector. A circuit that can detect both phase and frequency difference proves extremely useful because it significantly increases the acquisition range and lock speed of PLLs.

Unlike XOR gates and R-S latches, sequential phase/frequency detectors (PFDs) generate two outputs that are *not* complementary. Illustrated in Figure 42, the operation of a typical PFD is as follows. If the frequency of input A, ω_A, is less than that of input B, ω_B, then the PFD produces positive pulses at Q_A, while Q_B remains at zero. Conversely, if $\omega_A > \omega_B$, then positive pulses appear at Q_B while $Q_A = 0$. If $\omega_A = \omega_B$, then the circuit generates pulses at either Q_A or Q_B with a width equal to the phase difference between the two inputs. (Note that, in principle, Q_A and Q_B are never high simultaneously.) Thus, the average value of $Q_A - Q_B$ is an indication of the frequency or phase difference between A and B. The outputs Q_A and Q_B are usually called the "UP" and "DOWN" signals.

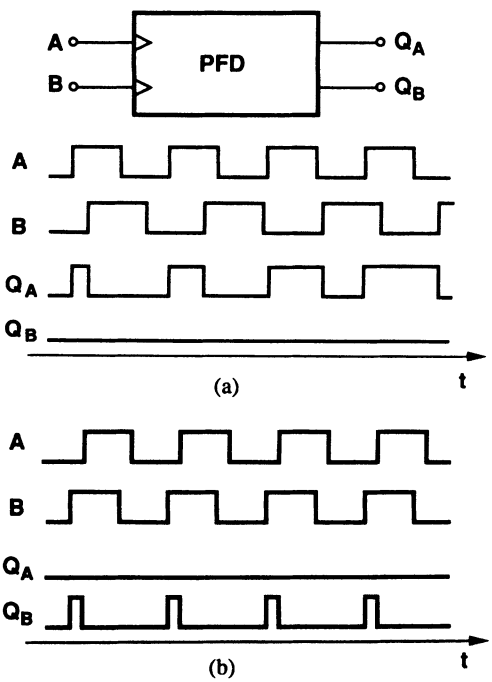

Fig. 42 Phase/frequency detector response with (a) $\omega_A < \omega_B$; (b) A lagging B.

To arrive at a circuit with the above behavior, we postulate that at least three logical states are required: $Q_A = Q_B = 0$; $Q_A = 0$, $Q_B = 1$; and $Q_A = 1$, $Q_B = 0$. Also, to avoid dependence of the output on the duty cycle of the inputs, the circuit should be implemented as an edge-triggered sequential machine. We assume the circuit can change state only on the rising transitions of A and B, and, for the sake of brevity, we will omit the adjective "rising" hereafter. Figure 43 shows a state diagram summarizing the operation. If the PFD is in the "ground" state, $Q_A = Q_B = 0$, then a transition on A takes it to State I, where $Q_A = 1$, $Q_B = 0$. The circuit remains in this state until a transition occurs on B, upon which the PFD returns to State 0. The switching sequence between States 0 and II is similar.

An important point in this state diagram is that if, for example, $\omega_A > \omega_B$, then there will be a time interval during which two transitions of A take place between two transitions of B. This ensures that, even if the PFD begins in State II, it will eventually leave that state and thereafter toggle between States 0 and I [16].

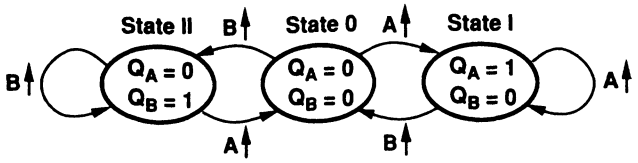

Fig. 43 PFD state diagram.

A possible implementation of the above PFD is shown in Figure 44 [16]. The circuit consists of two edge-triggered, resettable D flipflops with their D inputs connected to logical ONE. Signals A and B act as clock inputs of DFF_A and DFF_B, respectively. We note that if $Q_A = Q_B = 0$, a transition on A causes Q_A to go high. Subsequent transitions on A have no effect on Q_A, and when B goes high, the AND gate activates the reset of both flipflops. Thus, Q_A and Q_B are simultaneously high for a duration given by the total delay through the AND gate and the reset path of the flipflops. The implications of this overlap are explained later. Figure 45 shows the phase characteristic of the PFD.

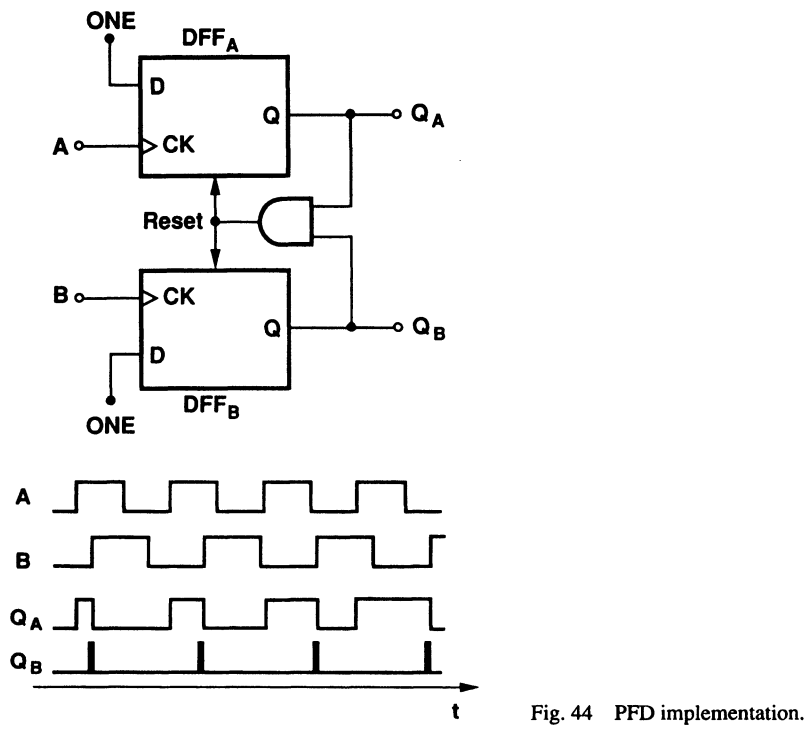

Fig. 44 PFD implementation.

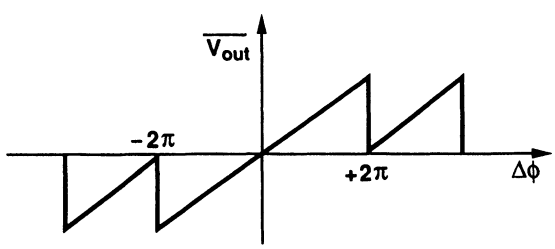

Fig. 45 PFD characteristic.

The D flipflops in Figure 44 may employ different topologies in bipolar and CMOS implementations. In bipolar technology, a standard master-slave configuration with an additional reset input can be used. In CMOS technology, a simple circuit such as that in Figure 46 [17] proves adequate. Note that the D input is "hidden" here.

Other implementations of PFDs are described in [18, 19].

The output of a PFD can be converted to dc in different ways. One approach is to sense the difference between the two outputs by means of a differential amplifier and apply the result to a low-pass filter. Alternatively, the outputs can drive a three-state "charge pump."

5.3 Charge Pumps

In the low-pass filters considered thus far (Figs. 15 and 17), the average value of the PD output is obtained by depositing charge onto a capacitor during each phase comparison and allowing the charge to decay afterwards. In a charge pump, on the other hand, there is negligible decay of charge between phase comparison instants, leading to interesting consequences.

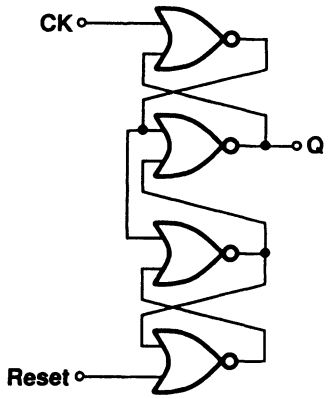

Fig. 46 Implementation of each *DFF* in Figure 44.

A three-state charge pump can be best studied in conjunction with a three-state phase/frequency detector (Fig. 47). The pump itself consists of two switched current sources driving a capacitor. (We assume herein that S_1 and I_1 are implemented with PMOSFETs and S_2 and I_2 with NMOSFETs.) Note that for a pulse of width T on Q_A, I_1 deposits a charge equal to IT on C_P. Thus, if $\omega_A > \omega_B$, or $\omega_A = \omega_B$ but A leads B, then positive charge accumulates on C_P steadily, yielding an *infinite* dc gain for the PFD. Similarly, if pulses appear on Q_B, I_2 removes charge from C_P on every phase comparison, driving V_{out} toward $-\infty$. In the third state, with $Q_A = Q_B = 0$, V_{out} remains constant. Since the steady-state gain is infinite, it is more meaningful to define the gain for one comparison instant, which is equal to $IT/(2\pi C_P)$.

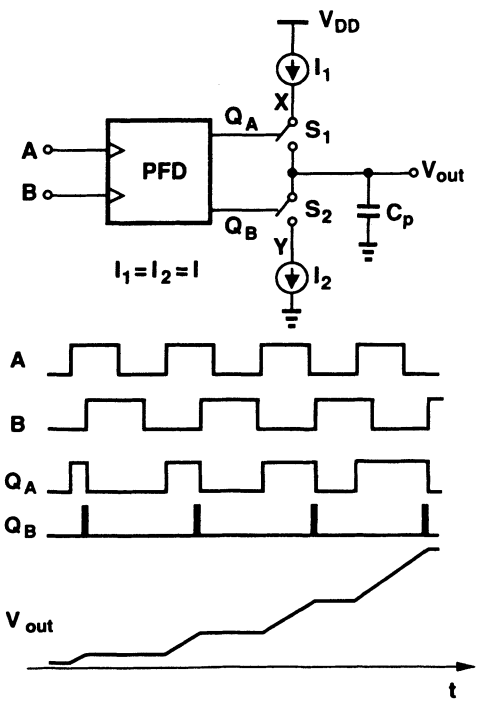

Fig. 47 PFD with charge pump.

An important conclusion to be drawn from the above observations is that, if offsets and mismatches are neglected, a PLL utilizing this arrangement locks such that the static phase difference between A and B is zero; even an infinitesimal phase error would result in an indefinite accumulation of charge on C_P.

The PFD/charge pump circuit of Figure 47 can potentially suffer from a "dead zone." To understand this effect, we let the phase difference between A and B approach zero and study the charge deposited on C_P. We also make two assumptions: 1) Q_A and Q_B exhibit relatively long transition times, for example, due to the capacitive loading of S_1 and S_2; 2) the delay from Q_A and Q_B through the AND gate and the reset path of the flipflops is small, i.e., when Q_A and Q_B exceed the threshold of the AND gate, the reset is immediately activated. The goal is to examine the increment in the charge deposited on C_P for an increment Δt in the delay between A and B (the small-signal gain).

Let us first consider a case where the delay between A and B is relatively large (Fig. 48a). We note that Q_A develops a full logical level and is reset when Q_B reaches the threshold of the AND gate. Thus, if the delay increments from T_1 to

$T_1 + \Delta t$, Q_A will be high for Δt seconds longer and the charge deposited on C_p will increase by $I\Delta t$. Now suppose, as shown in Figure 48b, $T_1 \approx 0$. What is the charge increment for a delay increment Δt? If B goes high Δt seconds after A does, then Q_B follows Q_A with the same delay. As soon as Q_B crosses the threshold of the AND gate, the reset signal is asserted, forcing Q_A and Q_B to return to zero. Thus, if Δt is small, Q_A does not reach a full logical level, failing to turn S_1 on or turning it on for an ill-defined length of time. In other words, the gain of the circuit drops as the delay between A and B becomes comparable with the transition time at Q_A and Q_B. This is illustrated in Figure 48c.

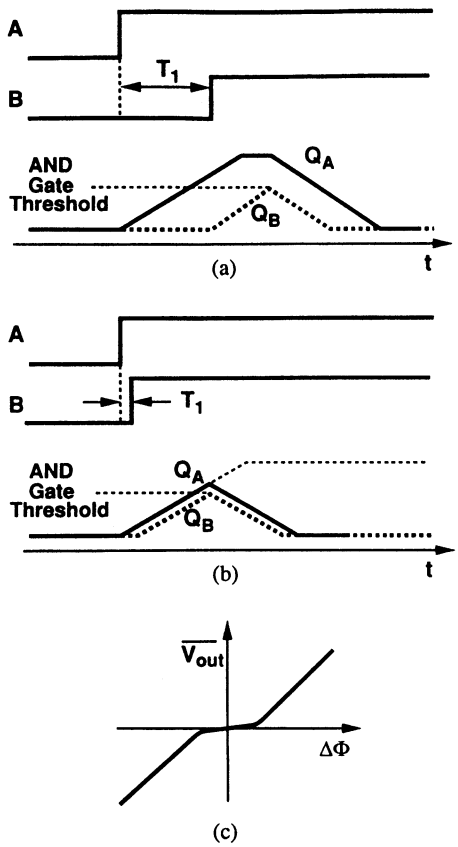

Fig. 48 Dead zone in a PFD with charge pump.

At this point, we must make two important observations. First, the dead zone is undesirable in a phase-locked system: If the phase difference between the input and output varies within the zone, the dc output of the charge pump does not change significantly and the loop fails to correct the resulting error. Consequently, a peak-to-peak jitter approximately equal to the width of the dead zone can arise in the output. Second, the PFD of Figure 44 is unlikely to suffer from a dead zone because its reset operation typically entails two or more gate delays (one due to the AND gate and one or more due to the flipflops). So long as the capacitive loading of S_1 and S_2 does not excessively slow down the output transitions, the reset delay ensures that Q_A and Q_B reach full logic levels. It is interesting to note that the dead zone existed in old implementations where the PFD output needed to drive an *external* charge pump and, hence, the capacitance associated with the IC pads and PC board traces. In those cases, additional inverters would be inserted at the output of the AND gate to increase the reset delay and eliminate the dead zone.

From the above discussion, we also infer that the dead zone disappears only if Q_A and Q_B can be simultaneously high for a sufficient amount of time. During this period, both S_1 and S_2 in Figure 47 are on, allowing the *difference* between I_1 and I_2 to vary the voltage stored on C_P. Since I_1 and I_2 typically have a few percent of mismatch, the output voltage varies even if the input phase difference is zero. Thus, a PLL employing this arrangement locks with a finite phase error so as to cancel the net charge deposited by I_1 and I_2 on C_p (Fig. 49). The important point is that the control voltage of the VCO is periodically disturbed, thereby modulating the VCO and introducing sidebands in the output spectrum.

Another error stems from mismatches between S_1 and S_2 in Figure 47. When these switches turn off, their charge-injection and feedthrough mismatch results in an error step at the output, changing the VCO frequency until the next phase comparison instant.

Our discussion thus far has assumed that I_1 and I_2 in Figure 47 are ideal. Since each current source requires a minimum voltage to maintain a relatively constant current, it is important that $V_{DD} - V_X$ and V_Y not drop below a certain

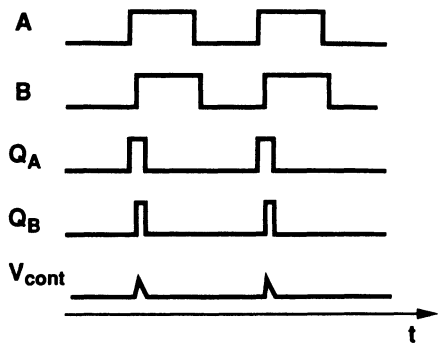

Fig. 49 Effect of mismatch in current sources of a charge pump.

level. If the extreme values of V_{cont} violate this condition, the current charging C_P varies and so does the overall gain [20], influencing the loop static and dynamic behavior.

Another related effect occurs when S_1 and S_2 are off: I_1 and I_2 pull nodes X and Y to V_{DD} and ground, respectively, causing charge-sharing between C_X, C_P, and C_Y when S_1 and S_2 turn on again (Fig. 50). If $V_{out} = V_{DD}/2$, $I_1 = I_2$, and $C_X = C_Y$, then V_{out} is not disturbed, but because V_{out} determines the VCO frequency, it is generally not equal to $V_{DD}/2$, thus experiencing a jump when S_1 and S_2 turn on. This effect is also periodic and introduces sidebands at the output, but it can be suppressed if nodes X and Y are bootstrapped to the voltage stored on the capacitor [7].

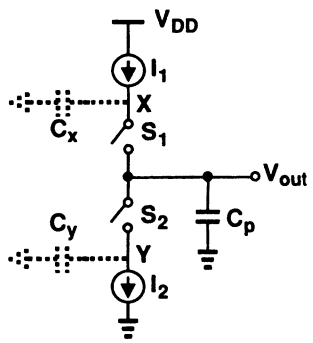

Fig. 50 Charge-sharing in a charge pump.

As noise immunity demands a differential control voltage for the VCO, the charge pump circuit of Figure 47 must be modified to provide a differential output. An additional advantage is that differential implementation alleviates the mismatch and charge-sharing problems as well. In a differential charge pump, the UP and DOWN signals activate only pull-down currents, and the pull-up currents are passive. Thus, when both UP and DOWN are low, a common-mode (CM) feedback circuit must counteract the pull-up currents to maintain a proper level.

Shown in Figure 51 is an example where differential pairs M_1-M_2 and M_3-M_4 are driven by the PFD and the network consisting of M_5-M_9 sets the output CM level at $V_{GS5} + V_{GS9}$. Because the CM level is momentarily disturbed at each phase comparison instant, it is important that CM transients not lead to *differential* settling components [21].

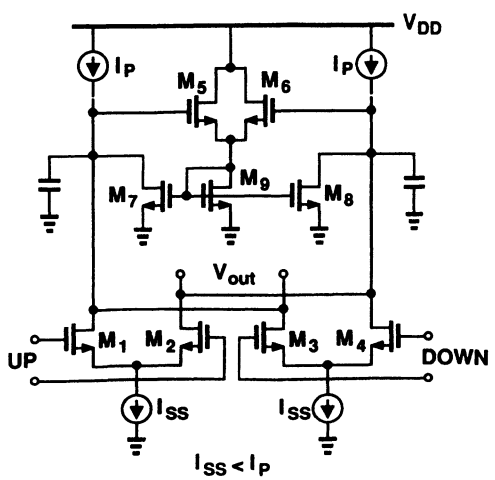

Fig. 51 Differential CMOS charge pump.

6. Charge-Pump Phase-Locked Loops

Charge-pump PLLs (CPPLLs) incorporate a PFD (or PD) and a charge pump (Fig. 52) instead of the combinational PD and the LPF in the generic architecture of Figure 11. As mentioned before, the combination of a PFD and a charge pump offers two important advantages over the XOR/LPF approach: 1) the capture range is only limited by the VCO output frequency range; 2) the static phase error is zero if mismatches and offsets are negligible. In this section, we study the characteristics of this type of PLL and make comparisons with the conventional type.

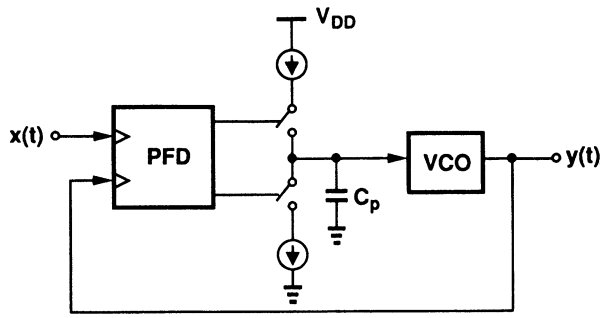

Fig. 52 Charge-pump PLL.

Charge pumps provide an infinite gain for a *static* phase difference at the input of the PFD. From another point of view, the response of a PFD/charge pump to a phase step is a linear ramp, indicating that the transfer function of the circuit contains a pole at *the origin.* With another such pole contributed by the VCO, a charge-pump PLL cannot remain stable. In fact, representing the transfer function of the PFD/charge pump with K_{PFD}/s, we note that the closed-loop transfer function of the PLL is

$$H(s) = \frac{\dfrac{K_{PFD}}{s}\dfrac{K_{VCO}}{s}}{1 + \dfrac{K_{PFD}}{s}\dfrac{K_{VCO}}{s}} \tag{53}$$

$$= \frac{K_{PFD}K_{VCO}}{s^2 + K_{PFD}K_{VCO}} \tag{54}$$

revealing two imaginary poles at $\omega = \pm j\sqrt{K_{PFD}K_{VCO}}$. To avoid instability, a zero must be added to the open-loop transfer function. This is in contrast to the case of a simple low-pass filter, where the loop is, in principle, stable even with no zero. The stabilizing zero in a CPPLL can be realized by placing a resistor in series with the charge-pump capacitor (Fig. 53).

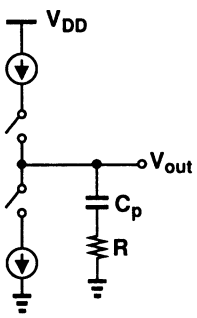

Fig. 53 Addition of a zero to a charge pump.

To perform a small-signal analysis, we note that the switching operation of the charge pump and the lack of a discharge path between phase comparison instants make the PLL a discrete-time system. However, if the loop bandwidth is much less than the input frequency, we can assume the state of the PLL changes by a small amount during each cycle of the input [20]. Using the "average" value of the discrete-time parameters, we can then study the loop as a continuous-time system [20].

Suppose the loop begins with a phase error $\phi_{in} - \phi_{out} = \phi_e$. Then, the average current charging the capacitor is given by $I\phi_e/(2\pi)$ and the average change in the control voltage of the VCO equals

$$V_{cont}(s) = \frac{I\phi_e}{2\pi}\left(R + \frac{1}{C_P s}\right) \tag{55}$$

Noting that $\Phi_{out}(s) = V_{cont}(s)K_{VCO}/s$, we obtain the following closed-loop transfer function:

$$H(s) = \frac{\frac{I}{2\pi C_P}(RC_P s + 1)K_{VCO}}{s^2 + \frac{I}{2\pi}K_{VCO}Rs + \frac{I}{2\pi C_P}K_{VCO}} \tag{56}$$

which has the same form as Eq. (36). Thus, the system is characterized by a zero at $\omega_z = -1/(RC_P)$ and

$$\omega_n = \sqrt{\frac{I}{2\pi C_P}K_{VCO}} \tag{57}$$

$$\zeta = \frac{R}{2}\sqrt{\frac{IC_P}{2\pi}K_{VCO}} \tag{58}$$

Note that ω_n is independent of R. If the loop includes frequency division, K_{VCO} must be divided by the division factor.

From Eq. (46), we note that the decay-time constant of the system is equal to $(\zeta\omega_n/2)^{-1} = (RIK_{VCO}/8)^{-1}$, a quantity independent of C_P.

In many applications, it is desirable to maximize the loop bandwidth, which is usually proportional to ω_n. While for a PLL with a sinusoidal PD (Section 3.3), ω_n and ζ cannot be maximized simultaneously, Eqs. (57) and (58) suggest that in a CPPLL both ω_n and ζ can be increased if I or K_{VCO} is increased. However, as the loop bandwidth becomes comparable with the input frequency, the continuous-time approximation used above breaks down, necessitating discrete-time analysis. Using such an analysis, Gardner has derived a stability limit [20] that can be reduced to

$$\omega_n^2 < \frac{\omega_{in}^2}{\pi(RC_P\omega_{in} + \pi)} \tag{59}$$

implying an upper bound on ω_n. This equation also indicates that R cannot be increased indefinitely [19]. (Typically, when the continuous-time approximation fails, the loop is unacceptably underdamped.)

In single-ended charge pumps, the resistor added in series with the capacitor can introduce "ripple" in the control voltage [20] even when the loop is locked. Since S_1 and S_2 turn on at every phase comparison instant, the mismatch between I_1 and I_2 flows through R, causing a step at the output. Furthermore, mismatch between overlap capacitance of S_1 and S_2 results in a net signal feedthrough to the output. Modulating the VCO frequency, this effect is especially undesirable in frequency synthesizers.

To suppress the ripple, a second capacitor can be connected from the output of the charge pump to ground. This modification introduces a third pole in the PLL, requiring further study of stability issues. Gardner provides criteria for the stability of such systems [20]. Note that fully differential charge pumps can reduce the magnitude of the ripple considerably.

The zero required in a charge-pump PLL can also be implemented using feedforward [22, 23]. This is accomplished by adding a fast signal path in parallel with the main charge pump. Illustrated in Figure 54 on page 31, this technique utilizes an auxiliary charge pump driving a wideband "dissipative" network, C_2 and R_2. The transfer function of the circuit is thus equal to

$$H_{CP}(s) = \frac{I_{P1}}{2\pi C_1 s} + \frac{I_{P2}R_2}{2\pi(R_2C_2s + 1)} \tag{60}$$

$$= \frac{(I_{P1}R_2C_2 + I_{P2}R_2C_1)s + I_{P1}}{2\pi C_1 s(R_2C_2s + 1)} \tag{61}$$

thereby providing a zero at

$$\omega_z = -\left(R_2C_2 + R_2C_1\frac{I_{P2}}{I_{P1}}\right)^{-1} \tag{62}$$

Note that the addition of the zero inevitably introduces a pole at $\omega_P = -1/(R_2C_2)$, making the PLL a third-order system. Moreover, the decay of the voltage across C_2 due to R_2 leads to ripple and, hence, frequency modulation of the VCO.

7. Noise in Phase-Locked Loops

Since PLLs operate on the phase of signals, they are susceptible to phase noise or jitter. Within the scope of this tutorial, we consider phase noise as a random component in the excess phase, as exemplified by $\phi_n(t)$ in Eq. (1). For the sake of brevity, we use the term *noise* to mean *phase noise*.

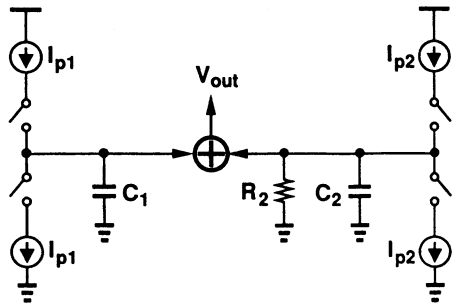

Fig. 54 Addition of zero by means of feedforward.

If the input signal or the building blocks of a PLL exhibit noise, then the output signal will also suffer from noise. In general, all the loop components, including the phase detector, the LPF, the VCO, and the frequency divider may contribute noise [24]. The goal is to understand how the spectrum of a given noise source is shaped as it propagates to the output.

In this tutorial, we examine two important cases: 1) the input signal contains noise, and 2) the VCO introduces noise; in each case, we find the transfer function from the noise source of interest to the PLL output. In monolithic implementations, the phase noise of the VCO is typically much more significant than that of other loop components.

7.1 Phase Noise at Input

Consider the PLL in Figure 55 where the input and output signals are $x(t) = A\sin[\omega_c t + \phi_{in}(t)]$ and $y(t) = B\sin[\omega_c t + \phi_{out}(t)]$. The transfer function $\Phi_{out}(s)/\Phi_{in}(s)$ is

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{63}$$

If the input (excess) phase, $\phi_{in}(t)$, does not vary with time, i.e., if the input to the PLL is a pure sinusoid, then $s = 0$ and $H(s) = 1$. Now, suppose $\phi_{in}(t)$ is varied so slowly that the denominator of Eq. (63) is still close to ω_n^2. Thus, $H(s)$ remains close to unity, indicating that the output phase (or frequency) follows the input phase (or frequency), a natural property of the PLL as a tracking system.

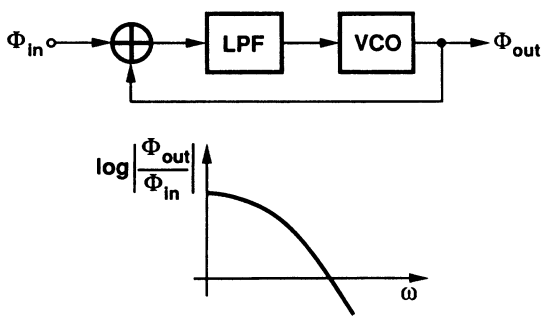

Fig. 55 Noise transfer function of a PLL from input to output.

What happens if $\phi_{in}(t)$ varies at an increasingly higher rate? Equation (63) shows that the output excess phase, $\phi_{out}(t)$, drops, eventually approaching zero and yielding $y(t) = B\sin\omega_c t$. In other words, for fast variations of the input excess phase or frequency, the PLL fails to track the input. Revisited in Section 9.1, this attribute of PLLs was the original reason for their widespread application in communications.

In summary, the input-phase noise spectrum of a PLL is shaped by the characteristic low-pass transfer function when it appears at the output. In order to minimize this noise, the loop bandwidth must be as small as possible, although it slows down the lock, limits the capture range, and degrades the stability.

7.2 Phase Noise of VCO

The phase noise of the VCO can be modeled as an additive component, ϕ_{VCO}, as shown in Figure 56. Assuming ϕ_{VCO} and ϕ_{in} are uncorrelated, we set ϕ_{in} to zero and compute the transfer function from ϕ_{VCO} to ϕ_{out}.[7] Note that $\phi_{in}(t) = 0$ means the *excess* phase of the input is zero, not the input signal itself; i.e., we must apply a strictly periodic signal at the input.

With $\phi_{in} = 0$ and a simple low-pass filter, we have

$$\frac{\Phi_{out}(s)}{\Phi_{VCO}(s)} = \frac{s(s + \omega_{LPF})}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{64}$$

[7]Superposition holds for the *power* of uncorrelated sources.

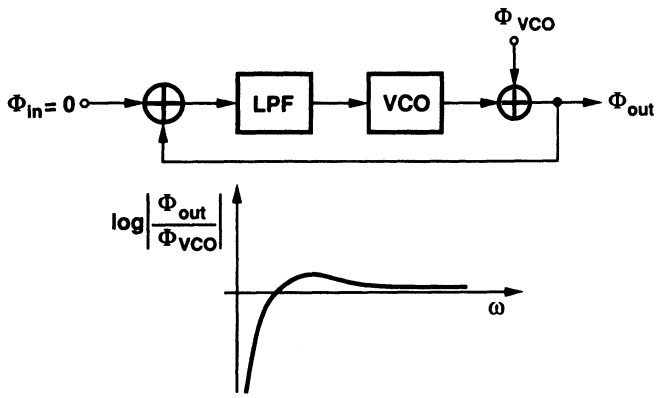

Fig. 56 Noise transfer function of a PLL from VCO to output.

As expected, this transfer function has the same poles as Eq. (63), but it also contains two zeros at $\omega_{z1} = 0$ and $\omega_{z2} = -\omega_{LPF}$, making the characteristic a *high-pass* filter.

The zero at the origin implies that, for slow variations in ϕ_{VCO}, ϕ_{out} is small. This is because, in lock, the phase variations in the VCO are converted to voltage by the PD and applied to the control input of the VCO so as to accumulate phase in the opposite direction. Since the VCO voltage/phase conversion has nearly infinite gain for a slowly varying V_{cont}, the negative feedback suppresses variations in the output phase.

From another perspective, the PLL can be simplified and redrawn as in Figures 57a and b. Because an ideal integrator placed in a negative feedback loop creates a "virtual" ground at its input, $\phi_{out} \approx 0$ for slow variations in ϕ_{VCO}.

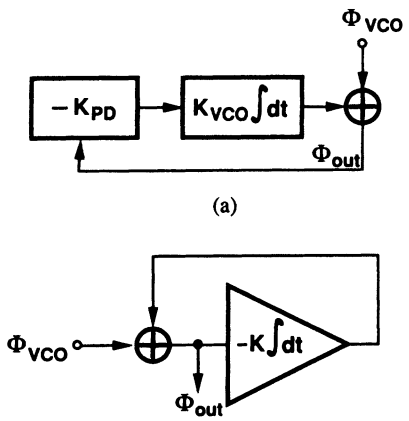

Fig. 57 Simplified model of PLL with VCO noise.

Now, suppose the rate of change of ϕ_{VCO} increases. Then, the magnitude of K_{VCO}/s and, hence, the loop gain decrease, allowing the virtual ground to experience significant variations. As the rate of change of ϕ_{in} approaches ω_{LPF}, the loop gain is reduced by the low-pass filter as well, an effect represented by the zero at $-\omega_{LPF}$.

From Eq. (64), we note that as $s \to \infty$, $\phi_{out} \to \phi_{VCO}$, which is to be expected because the feedback loop is essentially open for very fast changes in ϕ_{VCO}.

A common test of noise immunity in PLLs entails applying a small step to the power supply and finding the time required for the input-output phase difference to settle within a certain error band [18]. Since such a step predominantly affects the VCO output, we can use Eq. (64) to predict the circuit's behavior. For a phase step of height ϕ_1, the output assumes the following form:

$$\phi_{out}(t) = \phi_1 \left[\cos\sqrt{1-\zeta^2}\omega_n t + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin\sqrt{1-\zeta^2}\omega_n t \right] \times \exp(-\zeta\omega_n t) \tag{65}$$

Thus, the output initially jumps to ϕ_1 and subsequently decays to zero with a time constant $(\zeta\omega_n)^{-1}$. It is therefore desirable to maximize $\zeta\omega_n$ for fast recovery of the PLL.

From the above analysis, we conclude that to minimize the VCO phase-noise contribution, the loop bandwidth must be maximized, a requirement in conflict with that of the case where the PLL input contains noise. In applications where the

input has negligible noise (e.g., because it is derived from a crystal oscillator), the loop bandwidth is maximized to reduce both the VCO phase noise and the lock time.

8. Clock Recovery Circuits

As mentioned in Section 2.4, a clock recovery circuit produces a timing clock signal from a stream of binary data. In this section, we describe the design issues of CRCs and study various clock recovery techniques amenable to monolithic implementation.

8.1 Properties of NRZ Data

Binary data is commonly transmitted in the "nonreturn-to-zero" (NRZ) format. As shown in Figure 58a, in this format each bit has a duration of T_b ("bit period"), is equally likely to be ZERO or ONE, and is statistically independent of other bits. The quantity $r_b = 1/T_b$ is called the "bit rate" and measured in bits/s. The term "non-return-to-zero" distinguishes this type from another one called the "return-to-zero" (RZ) format, in which the signal goes to zero between consecutive bits (Fig. 58b). Since for a given bit rate, RZ data contains more transitions than NRZ data, the latter is preferable where channel or circuit bandwidth is costly. Note that, in general, data must be treated as a random waveform (with certain known statistical properties).

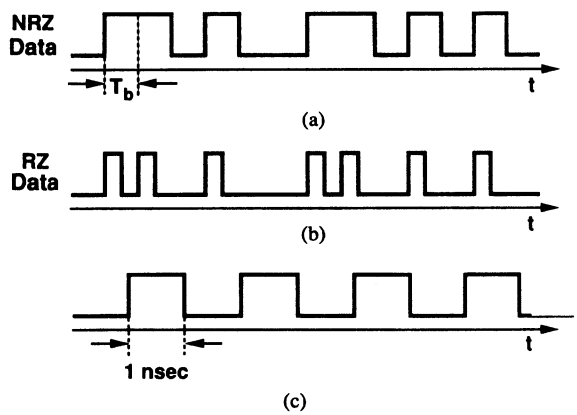

Fig. 58 (a) NRZ data; (b) RZ data; (c) fastest NRZ waveform with r_b = 1 Gb/s.

NRZ data has two attributes that make the task of clock recovery difficult. First, the data may exhibit long sequences of consecutive ONEs or ZEROs, demanding the CRC to "remember" the bit rate during such a period. This means that, in the absence of data transitions, the CRC should not only continue to produce the clock, but also incur negligible drift in the clock frequency. We return to this issue later.

Second, the spectrum of NRZ data has nulls at frequencies that are integer multiples of the bit rate; for example, if the data rate is 1 Gb/s, the spectrum has no energy at 1 GHz. To understand why, we note that the fastest waveform for a 1-Gb/s stream of data is obtained by alternating between ONE and ZERO every 1 ns (Fig. 58c). The result is a 500-MHz square wave, with all the even-order harmonics absent. From another point of view, if an NRZ sequence with rate r_b is multiplied by $A\sin(2\pi m r_b t)$, the result has a zero average for all integers m, indicating that the waveform contains no frequency components at mr_b.

It is also helpful to know the shape of the NRZ data spectrum. Since the autocorrelation function of a random binary sequence is [25]

$$R_x(\tau) = 1 - \frac{|\tau|}{T_b} \qquad |\tau| < T_b \tag{66}$$

$$= 0 \qquad |\tau| < T_b \tag{67}$$

the power spectral density equals

$$P_x(\omega) = T_b \left[\frac{\sin(\omega T_b/2)}{\omega T_b/2}\right]^2 \tag{68}$$

Plotted in Figure 59, this function vanishes at $\omega = 2m\pi/T_b$. In contrast, RZ data has finite power at such frequencies.

Due to the lack of a spectral component at the bit rate in the NRZ format, a clock recovery circuit may lock to spurious signals or simply not lock at all. Thus, NRZ data usually undergoes a nonlinear operation at the front end of the circuit so as to create a frequency component at r_b. A common approach is to detect each data transition and generate a corresponding pulse.

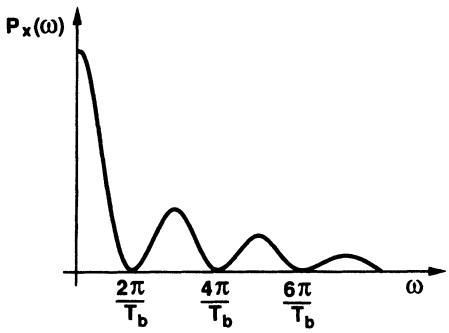

Fig. 59 Spectrum of NRZ data.

8.2 Edge Detection

Illustrated in Figure 60a, edge detection requires a means of sensing both positive and negative data transitions. In Figure 60b, an XOR gate with a delayed input performs this operation, whereas in Figure 60c, a differentiator produces impulses corresponding to each transition, and a squaring circuit or a full-wave rectifier converts the negative impulses to positive ones.

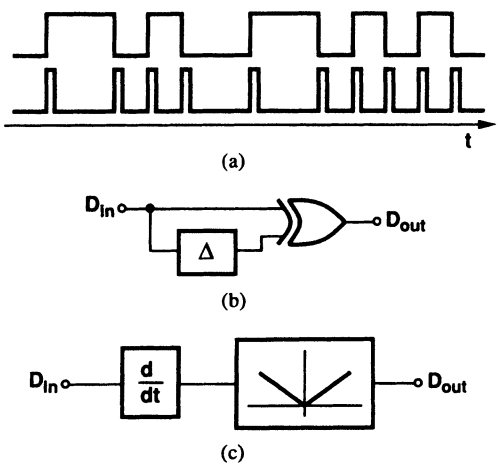

Fig. 60 Edge detection of NRZ data.

A third method of edge detection employs a flipflop operating on both rising and falling edges [26]. To understand this technique, we first note that, in a phase-locked CRC, the edge-detected data is multiplied by the output of a VCO (Fig. 61a). In effect, the data transition impulses "sample" points on the VCO output. This can also be accomplished using a master-slave flipflop consisting of two D latches: the data pulses drive the clock input while the VCO output is sensed by the D input (Fig. 61b). Since in this configuration the VCO output is sampled on either rising or falling transitions of the data, we modify the circuit such that both latches sample X_{VCO}, but on opposite transitions of D_{in}. Shown in Figure 61c, the resulting circuit samples the VCO output on every data transition and is therefore functionally equivalent to that in Figure 61a. We call this circuit a double-edge-triggered flipflop.

8.3 Clock Recovery Architectures

From the above observations, we note that clock recovery consists of two basic functions: 1) edge detection; 2) generation of a periodic output that settles to the input data rate but has negligible drift when some data transitions are absent. Illustrated in Figure 62a is a conceptual realization of these functions, where a high-Q oscillator is "synchronized" with the input transitions and oscillates freely in their absence. The synchronization can be achieved by means of phase-locking.

Figure 62b shows how a simple PLL can be used along with edge detection to perform clock recovery. First, suppose the input data is periodic with a frequency $1/T_b$. (The unit of $1/T_b$ is hertz rather than rad/s.) Then, the edge detector simply doubles the frequency, allowing the VCO to lock to $1/(2T_b)$. Now, assume some transitions are absent. During such an interval, the output of the multiplier is zero and the voltage stored in the low-pass filter decays, thereby making the VCO frequency drift. To minimize this effect, the time constant of the LPF must be sufficiently larger than the maximum allowable interval between consecutive transitions,[8] thereby resulting in a small loop bandwidth and, hence, a narrow capture range.

It follows from the above discussion that a PLL used for clock recovery must also employ frequency detection to ensure locking to the input despite process and temperature variations. This may suggest replacing the multiplier with the

[8]Most communication systems guarantee a certain upper bound on this interval by encoding the data.

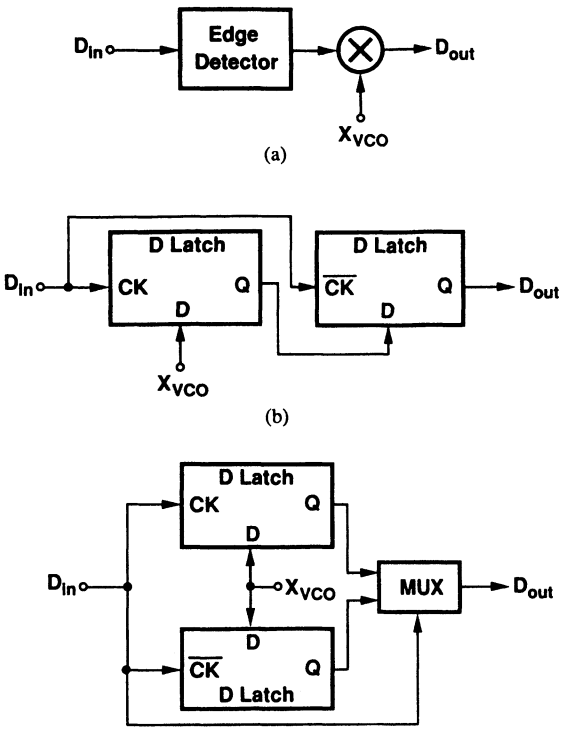

Fig. 61 Edge detection and sampling of NRZ data.

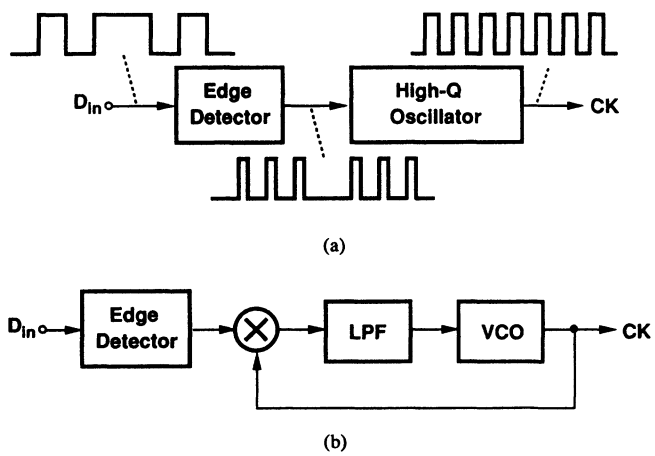

Fig. 62 (a) Conceptual realization of a CRC; (b) phase-locked CRC.

three-state PFD of Figure 42. However, the latter circuit produces an incorrect output if either of its input signals exhibits missing transitions. As depicted in Figure 63, in the absence of transitions on the main input, the PFD interprets the VCO frequency to be higher than the input frequency, driving the control voltage in such a direction as to correct the apparent difference. This occurs even if the VCO frequency is initially equal to the input data rate. Thus, the choice of phase and frequency detectors for random binary data requires careful examination of their response when some transitions are absent.

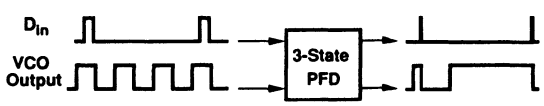

Fig. 63 Response of a three-state PFD to random data.

A clock recovery architecture that has been implemented in both analog and digital domains is the "quadricorrelator," introduced by Richman [27] and modified by Bellisio [28]. We first consider an analog representation of the architecture to describe its underlying principles. Shown in Figure 64, and bearing some resemblance to that in Figure 25, the quadricorrelator follows the edge detector with a combination of three loops sharing the same VCO. Loops I and II perform frequency detection and Loop III, phase detection. Note that the VCO generates quadrature outputs. The circuit operates as follows. Suppose the

edge detector produces a frequency component at ω_1 while the VCO oscillates at ω_2. Mixing the VCO outputs with $\sin \omega_1 t$ and low-pass filtering the results, we obtain quadrature beat signals $\sin(\omega_1 - \omega_2)t$ and $\cos(\omega_1 - \omega_2)t$. Next, the latter signal is differentiated and mixed with the former, yielding $(\omega_1 - \omega_2)\cos^2(\omega_1 - \omega_2)t$ at node P. Representing both the polarity and the magnitude of the difference between ω_1 and ω_2, the average value of this signal drives the VCO with negative feedback so as to bring ω_2 closer to ω_1. As $|\omega_2 - \omega_1|$ drops, Loop III—a simple PLL—begins to generate an asymmetric signal at node M, assisting the lock process. For $\omega_2 \approx \omega_1$, the dc feedback signal produced by Loops I and II approaches zero and Loop III dominates, locking the VCO output to the input data.

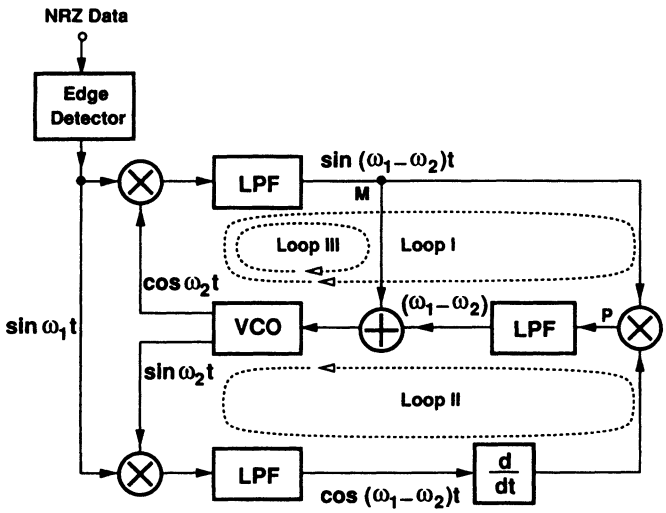

Fig. 64 Quadricorrelator.

The use of frequency detection in the quadricorrelator makes the capture range independent of the (locked) loop bandwidth, allowing a small cut-off frequency in the LPF of Loop III so as to minimize the VCO drift between data transitions.[9] Nevertheless, because the frequency detection circuit can respond to noise and spurious components, it is preferable to disable Loops I and II once phase lock is attained.

Further analysis of the quadricorrelator is given in [27, 29].

A drawback of the quadricorrelator in discrete technologies was that it required quadrature outputs of the VCO, a problem solved by passing the VCO output through a delay line to shift the phase by 90°. The dependence of such a delay on frequency, temperature, and component values made the design difficult. In monolithic implementations, on the other hand, differential ring oscillators with an even number of stages easily provide quadrature outputs (Section 5.1).

The quadricorrelator can also be realized in digital form. Recall from Figure 61 that the combination of an edge detector and a mixer can be replaced with a double-edge-triggered flipflop. Thus, the architecture of Figure 64 can be "digitalized" as shown in Figure 65, which is notably similar to that in [26].

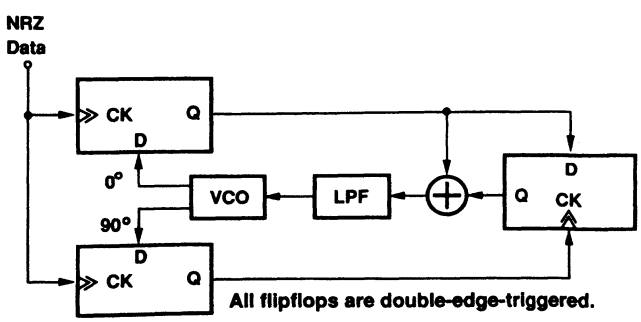

Fig. 65 Digital implementation of quadricorrelator.

Other types of phase and frequency detectors for NRZ data are described in [30, 31].

[9]It may be necessary to follow the mixer in Loop III with *two* LPFs: one with a wide band in Loop I and another with a narrow band in Loop III.

9. APPLICATIONS

9.1 Noise and Jitter Suppression in Communications

A common situation in communication systems is that a narrowband signal is corrupted by noise. This occurs, for example, in satellite transceivers where weak signals buried in noise must be detected by coherent demodulation. In order to achieve a high signal-to-noise ratio, the noise components around the carrier, ω_c, must be suppressed, implying the need for a narrowband filter. However, in most applications the required filter bandwidth is several orders of magnitude smaller than ω_c, thereby demanding filter Qs of greater than 1000.

A PLL can operate as a narrowband filter with an extremely high Q (Fig. 66). Recall that the input/output phase (or frequency) transfer function of a (continuous-time) PLL is that of a low-pass filter whose bandwidth is *independent* of the input frequency. Making the bandwidth of the PLL sufficiently small can therefore result in a very high equivalent Q. From another point of view, the PLL takes the average of the input frequency over a great many cycles, thus suppressing variations therein.

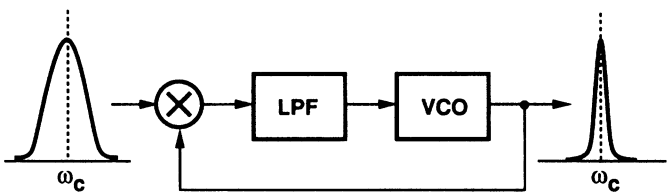

Fig. 66 PLL as a narrowband filter.

As mentioned in Section 7, with a small loop bandwidth, the VCO becomes the dominant source of output noise. Thus, in this application fully monolithic VCOs may not achieve sufficiently low phase noise, and external resonant devices such as inductors may be required.

In digital communications, transmitted or retrieved data often suffer from timing jitter, a problem similar to that illustrated in Figure 1. In order to lower the jitter, the data can be "regenerated" (or "retimed") with the aid of a phase-locked clock recovery circuit. Depicted in Figure 67, regeneration occurs when the data waveform is sampled at its peaks, farthest from zero-crossing points. Subsequent amplification then produces a logical ONE or ZERO according to the polarity of the sampled value.

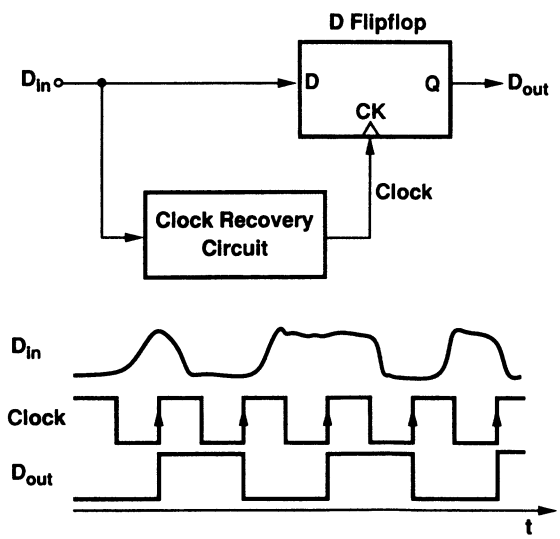

Fig. 67 Jitter suppression by data regeneration.

To appreciate the efficiency and robustness of this approach, we make two observations. First, as with the case shown in Figure 66, the phase-locked CRC rejects most of the noise that accompanies the data, generating a low-jitter clock. Second, peak sampling produces a correct output even if the data zero crossings deviate from their ideal instants by almost half a bit period (Fig. 68). Heavily deteriorated data can thus be recovered.

An important issue in clock and data recovery is "jitter peaking." In most PLLs, the input/output transfer function contains a zero, typically exhibiting a peaking in the frequency response. As a result, some high-frequency jitter components are actually amplified as they appear at the output. This issue is further discussed in [32, 33].

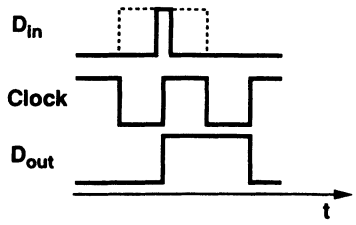

Fig. 68 Jitter margin for correct sampling of data.

9.2 Skew Suppression in Digital Systems

As described in Section 2.2, the interface between off-chip clocks and high-speed digital ICs typically introduces significant skew. This originates from the finite delay of the on-chip clock buffers used to drive the device and interconnect capacitance. Considering the delay as a phase shift, we postulate that its effect can be reduced if the buffers are embedded in a PLL with large loop gain. Figure 69 is an example of the on-chip buffered signal, CK_C, phase-locked to the off-chip clock, CK_S. With a three-state PFD and charge pump, the buffer delay is, in principle, completely cancelled.

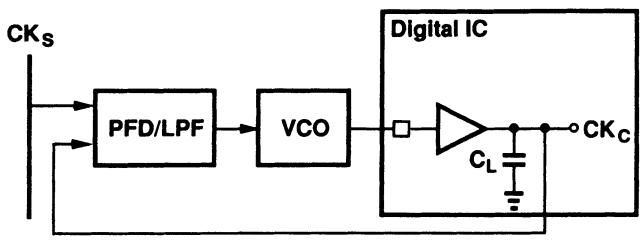

Fig. 69 Skew reduction using a PLL.

The above technique can be utilized more extensively in large clock networks. For example, if a processor clock is distributed on an "H" tree, the signal at the end of one of the branches can be locked to the input so as to null the delay associated with intermediate wires and local buffers [34].

An important issue in PLLs used in low-power systems is the lock time. A low-power processor is frequently powered down and up, requiring the PLL to perform phase alignment quickly. The load capacitance seen by the clock buffer changes when the processor is powered up. Thus, even if the PLL is always on, it still experiences a large transient. Another difficulty is that the lock time is a function of loop parameters and, hence, varies considerably with process and temperature.

9.3 Frequency Synthesis in RF Transceivers

RF systems usually require a high-frequency local oscillator (LO) whose frequency can be changed in small, precise steps. The ability of PLLs to multiply frequencies makes them attractive for synthesizing frequencies.

Figure 70 shows an example of a phase-locked synthesizer. The goal is to generate an output frequency that can be varied from 900 MHz to 925.4 MHz in steps of 200 kHz, covering 128 channels. The frequency divider in the loop is designed such that its division ratio is $M = NP + S$, where $NP = 4500$ and S can be programmed from zero to 127 by the "channel select" input. Thus, if $f_{REF} = 200$ kHz, then f_{out} can be varied from $f_{min} = (4500 + 0) \times 200$ kHz $= 900$ MHz to $f_{max} = (4500 + 127) \times 200$ kHz $= 925.4$ MHz.

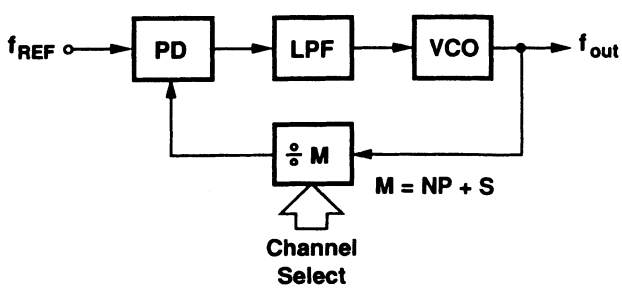

Fig. 70 Phase-locked synthesizer.

RF synthesizers typically impose severe constraints on the output phase noise and sidebands. In the example of Figure 70, phase noise is contributed mostly by the VCO because f_{REF} is provided by a low-noise crystal oscillator. Thus, it is desirable to maximize the loop bandwidth. However, as explained in Section 4.8, the second harmonic of f_{REF} at the PD output creates sidebands in the VCO, and can be suppressed only by lowering ω_{LPF}. This leads to a trade-off between phase noise and sideband magnitudes, making the architecture a viable choice only if the VCO phase noise is sufficiently small.

The lock time of RF synthesizers is also an important parameter. In "frequency-hopped" systems, for example, the channel is required to change in a short amount of time, a constraint in conflict with the small loop bandwidth needed to suppress the sidebands. Many architectures and circuit techniques have been devised to resolve these issues [35].

10. CONCLUSION

The concept of phase locking has proved essential in today's electronic systems. The ability of PLLs to control phase and frequency with extremely high precision provides efficient solutions to various design problems in communications, RF and wireless applications, disk drive electronics, and digital systems.

11. REFERENCES

[1] F. M. Gardner. *Phaselock Techniques*, Second Edition. New York: Wiley & Sons, 1979.

[2] A. B. Grebene, "The monolithic phase locked loop—A versatile building block," *IEEE Spectrum*, vol. 8, pp. 38–49, March 1971.

[3] R. E. Best. *Phase-Locked Loops*, Second Edition. New York: McGraw-Hill, 1993.

[4] G. S. Moschytz, "Miniaturized RC filters using phase-locked loop," *Bell Syst. Tech. J.*, vol. 44, pp. 823–870, May 1965.

[5] W. F. Egan. *Frequency Synthesis by Phase Lock*. New York: Wiley & Sons, 1981.

[6] M. Bazes, "A novel precision MOS synchronous delay line," *IEEE J. Solid-State Circuits*, vol. 20, pp. 1265–1271, December 1985.

[7] M. G. Johnson and E. L. Hudson, "A variable delay line PLL for CPU-coprocessor synchronization," *IEEE J. Solid-State Circuits*, vol. 23, pp. 1218–1223, October 1988.

[8] J. Sonntag and R. Leonowich, "A monolithic CMOS 10 MHz DPLL for burst-mode data retiming," *ISSCC Dig. Tech. Papers*, pp. 194–195, San Francisco, CA, February 1990.

[9] B. van der Pol, "The nonlinear theory of electric oscillators," *Proc. IRE*, vol. 22, pp. 1051–1086, September 1934.

[10] A. Borys, "Elementary deterministic theories of frequency and amplitude stability in feedback oscillators," *IEEE Trans.,Circuits Syst.*, vol. 34, pp. 254–258, March 1987.

[11] B. Razavi, "Analysis, modeling, and simulation of phase noise in monolithic voltage-controlled oscillators," *Proc. CICC*, pp. 323–326, Santa Clara, CA, 1995.

[12] B. Lai and R. C. Walker, "A monolithic 622 Mb/sec clock extraction and data retiming circuit," *ISSCC Dig. Tech. Papers*, pp. 144–145, San Francisco, CA, February 1991.

[13] S. K. Enam and A. A. Abidi, "NMOS ICs for clock and data regeneration in gigabit-per-second optical-fiber receivers," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1763–1774, December 1992.

[14] A. Pottbacker and U. Langmann, "An 8 GHz silicon bipolar clock recovery and data regenerator IC," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1572–1576, December 1994.

[15] B. Razavi and J. Sung, "A 2.5-Gb/sec 15-mW BiCMOS clock recovery circuit," *Symp. VLSI Circuits Dig. Tech. Papers*, pp. 83–84, Kyoto, Japan, 1995.

[16] C. A. Sharpe, "A 3-state phase detector can improve your next PLL design," *EDN*, pp. 55–59, September 20, 1976.

[17] I. Shahriary et al., "GaAs monolithic phase/frequency discriminator," *IEEE GaAs IC Symp. Dig. Tech. Papers*, pp. 183–186, 1985.

[18] I. A. Young, J. K. Greason, and K. L. Wong, "A PLL clock generator with 5 to 110 MHz of lock range for microprocessors," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1599–1607, November 1992.

[19] D. K. Jeong et al., "Design of PLL-based clock generation circuits," *IEEE J. Solid-State Circuits*, vol. 22, pp. 255–261, April 1987.

[20] F. M. Gardner, "Charge-pump phase-locked loops," *IEEE Trans. Comm.*, vol. COM-28, pp. 1849–1858, November 1980.

[21] B. Razavi. *Principles of Data Conversion System Design*. Piscataway, NJ: IEEE Press, 1995.

[22] D. Mijuskovic et al., "Cell-based fully integrated CMOS frequency synthesizers," *IEEE J. Solid-State Circuits*, vol. 29, pp. 271–279, March 1994.

[23] I. Novof et al., "Fully-integrated CMOS phase-locked loop with 15 to 240 MHz locking range and ±50 psec jitter," *ISSCC Dig. Tech. Papers*, pp. 112–113, San Francisco, CA, February 1995.

[24] V. F. Kroupa, "Noise properties of PLL systems," *IEEE Trans. Comm.*, vol. COM-30, pp. 2244–2252, October 1982.

[25] S. K. Shanmugam. *Digital and Analog Communication Systems*. New York: Wiley & Sons, 1979.

[26] A. Pottbacker, U. Langmann, and H. U. Schreiber, "A Si bipolar phase and frequency detector IC for clock extraction up to 8 Gb/s," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1747–1751, December 1992.

[27] D. Richman, "Color-carrier reference phase synchronization accuracy in NTSC color television," *Proc. IRE*, vol. 42, pp. 106–133, January 1954.

[28] J. A. Bellisio, "A new phase-locked loop timing recovery method for digital regenerators," *IEEE Int. Communications Conf. Rec.*, vol. 1, June 1976, pp. 10-17–10-20.

[29] F. M. Gardner, "Properties of frequency difference detectors," *IEEE Trans. Comm.*, vol. COM-33, pp. 131–138, February 1985.

[30] D. G. Messerschmitt, "Frequency detectors for PLL acquisition in timing and carrier recovery," *IEEE Trans. Comm.*, vol. COM-27, pp. 1288–1295, September 1979.

[31] C. R. Hogge, "A self-correcting clock recovery circuit," *IEEE J. Lightwave Technology*, vol. LT-3, pp. 1312–1314, December 1985.

[32] L. De Vito et al., "A 52 MHz and 155 MHz clock recovery PLL," *ISSCC Dig. Tech. Papers*, pp. 142–143, San Francisco, CA, February 1991.

[33] T. H. Lee and J. F. Bulzacchelli, "A 155-MHz clock recovery delay- and phase-locked loop," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1736–1746, December 1992.

[34] J. Alvarez et al., "A wide-bandwidth low-voltage PLL for PowerPC microprocessors," *IEEE J. Solid-State Circuits*, vol. 30, pp. 383–391, April 1995.

[35] J. A. Crawford. *Frequency Synthesizer Design Handbook*. Norwood, MA: Artech House, 1994.