# Phase-Domain All-Digital Phase-Locked Loop

Robert Bogdan Staszewski and Poras T. Balsara

***Abstract*— A fully digital frequency synthesizer for RF wireless applications has recently been proposed. At its foundation lies a digitally controlled oscillator that deliberately avoids any analog tuning controls. When implemented in a digital deep-submicrometer CMOS process, the proposed architecture appears more advantageous over conventional charge-pump-based phase-locked loops (PLLs), since it exploits signal processing capabilities of digital circuits and avoids relying on the fine voltage resolution of analog circuits. An actual implementation of an all-digital PLL (ADPLL)-based local oscillator and transmitter used in a commercial 0.13-$\mu$m CMOS single-chip Bluetooth radio has recently been disclosed. The conventional phase/frequency detector, charge pump and *RC* loop filter are replaced by a time-to-digital converter and a simple digital loop filter. Due to the lack of the correlational phase detection mechanism, the loop does not contribute to the reference spurs. The measured close-in phase noise of $-86$ dBc/Hz is adequate even for Global System for Mobile communications (GSM) applications. In this paper, we present the mathematical description and operational details of the phase-domain ADPLL.**

***Index Terms*—All-digital phase-locked loop (ADPLL), Bluetooth, Global System for Mobile communications (GSM), phase domain, PLL, synchronous, wireless.**

## I. Introduction

WITH the explosive growth of the wireless communications industry worldwide, the need has arisen to reduce cost and power consumption of mobile stations. The use of deep-submicrometer CMOS processes allows for an unprecedented degree of scaling and integration in digital circuitry, but complicates the implementation of traditional RF and analog circuits. Consequently, a new research focuses on finding digital architectural solutions to these integration problems.

The frequency synthesizer is a key block used for both up-conversion and down-conversion of radio signals and has been traditionally based on a charge-pump PLL, which is not easily amenable to integration. Recently, a *digitally controlled oscillator* (DCO), which deliberately avoids any analog tuning voltage controls, was first ever presented in [2] for RF wireless applications. This allows for its loop control circuitry to be implemented in a fully digital manner as first proposed in [1] and then demonstrated as a novel digital-synchronous phase-domain all-digital PLL (ADPLL) in a commercial 0.13-$\mu$m CMOS single-chip Bluetooth radio [3].

Per PLL classification in [4], the proposed frequency synthesizer is not a classical *digital PLL* (DPLL), which actually is considered a semi-analog circuit, but an ADPLL with all building blocks defined as digital at the input/output level. Similar to a flip-flop, which is a cornerstone of digital circuits, the DCO is built as an application-specific integrated circuit (ASIC) cell, whose internals are analog, but the analog nature does not propagate beyond its boundaries [2]. So far, there are no other comparable architectures reported in the literature.

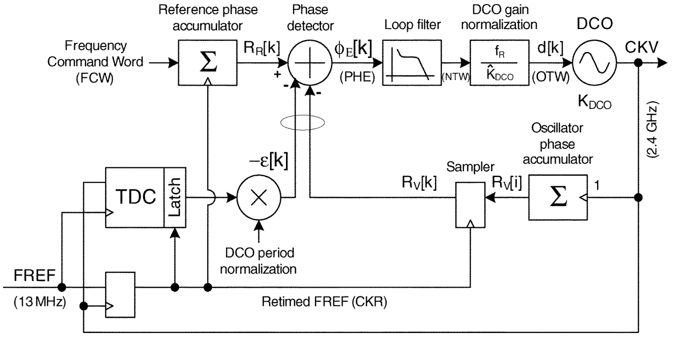

Fig. 1. ADPLL-based RF frequency synthesizer.

In this paper, we present a mathematical description and operational details of the phase-domain ADPLL. The phase-domain operation is motivated by an observation [5], that, since the reference phase and oscillator phase are in a linear form, their difference produced by the phase detector is also linear with no spurs and the loop filter would not be needed. This is in contrast with conventional charge-pump-based PLLs, whose phase detection operation is correlational and generates significant amount of spurs that require a strong loop filter that degrades the transients and limits the switching time.

## II. ADPLL

Fig. 1 shows a block diagram of the ADPLL-based frequency synthesizer. It operates in a digitally synchronous fixed-point phase domain, which was recently proposed in [3]. The variable phase signal $R_V[i]$ is determined by counting the number of rising clock transitions of the DCO oscillator clock. The reference phase signal $R_R[k]$ is obtained by accumulating the *frequency command word* (FCW) with every rising edge of the retimed *frequency reference* (FREF) clock. The sampled variable phase $R_V[k]$ is subtracted from the reference phase in a synchronous arithmetic phase detector. The digital phase error is conditioned by a simple digital loop filter and then normalized by the DCO gain, $K_{\mathrm{DCO}}$. The $K_{\mathrm{DCO}}$ normalization is needed to precisely establish the loop bandwidth and to perform a direct transmit frequency modulation, as described in [6]. The FREF input is resampled by the RF oscillator clock, and the resulting *retimed clock* (CKR) is used throughout the system. This ensures that the massive digital logic is clocked after the quiet interval of the phase error detection by the *time-to-digital converter* (TDC). A detailed description follows below.

Manuscript received January 10, 2004; revised July 5, 2004. This paper was recommended by Associate Editor T. Ueta.

R. B. Staszewski is with the Wireless Analog Technology Center, Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: b-staszewski@ti.com).

P. T. Balsara is with the Center for Integrated Circuits and Systems, University of Texas at Dallas, Richardson, TX 75083 USA.

Digital Object Identifier 10.1109/TCSII.2004.842067

## III. Phase-Domain Operation

Let us define the actual clock period of the variable oscillator (DCO or voltage-controlled oscillator, in general) output (CKV) as $T_V$ and the clock period of the FREF as $T_R$. Let us assume that the oscillator runs appreciably faster than the available reference clock, $T_V \ll T_R$, which is the case in RF synthesizers, where the generated multigigahertz RF carrier frequency is of orders of magnitude higher than a typical 10–40-MHz crystal reference. Let us further assume, in order to simplify the initial analysis, that the actual clock periods are constant or time invariant.

The CKV and FREF clock transition timestamps $t_V$ and $t_R$, respectively, are governed by the following equations:

$$t_V = i \cdot T_V \tag{1}$$

$$t_R = k \cdot T_R + t_0 \tag{2}$$

where, $i = 1, 2, \ldots$ and $k = 1, 2, \ldots$ are the CKV and FREF clock transition index numbers, respectively, and $t_0$ is some initial time offset between the two clocks, which is absorbed into the FREF clock.

It is convenient in practice to normalize the transition timestamps in terms of actual $T_V$ units , referred to as *unit intervals* (UI), since it is easy to observe and operate on the actual CKV clock events. Let us define dimensionless variable and reference "phase"

$$\theta_V \equiv \frac{t_V}{T_V} \tag{3}$$

$$\theta_R \equiv \frac{t_R}{T_V}. \tag{4}$$

The term $\theta_V$ is only defined at CKV transitions and indexed by $i$. Similarly, $\theta_R$ is only defined at FREF transitions and indexed by $k$. This results in

$$\theta_V[i] = i \tag{5}$$

$$\theta_R[k] = k \cdot \frac{T_R}{T_V} + \frac{t_0}{T_V} = k \cdot N + \theta_0. \tag{6}$$

The normalized transition timestamps $\theta_V[i]$ of the variable clock, CKV, could be estimated by accumulating the number of significant (rising or falling) edge clock transitions

$$R_V(i \cdot T_V) \equiv R_V[i] = \sum_{l=0}^{i} 1. \tag{7}$$

Without the FREF retiming (described in Section IV), the normalized transition timestamps $\theta_R[k]$ of the FREF clock, could be obtained by accumulating the FCW on every significant edge of the FREF clock

$$R_R(k \cdot T_R) \equiv R_R[k] = \sum_{l=0}^{k} \mathrm{FCW}. \tag{8}$$

FCW is formally defined as the frequency division ratio of the *expected* variable frequency to the reference frequency

$$\mathrm{FCW} \equiv \frac{\mathcal{E}(f_V)}{f_R}. \tag{9}$$

The reference frequency is usually of excellent long-term accuracy, at least as compared to the variable oscillator. For this reason, we do not use the expectation operator on $f_R$.

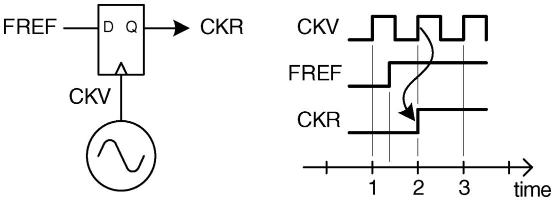

Fig. 2. Concept of synchronizing the clock domains by retiming the FREF.

FCW control is generally expressed as being comprised of an integer ($N_i$) and fractional ($N_f$) parts

$$\mathrm{FCW} = N = N_i + N_f. \tag{10}$$

The PLL achieves, in a steady-state condition, a zero or constant averaged phase difference between the variable $\theta_V[i]$ and the reference $\theta_R[k]$ normalized timestamps. Attempt to formulate the phase error as this unitless phase difference $\phi_E = \theta_R - \theta_V$ would be unsuccessful due to the nonalignment of the time samples. This will be addressed in Section IV.

The additional benefit of operating the PLL with phase-domain signals is to alleviate the need for the frequency-detection function within the phase detector. This allows to operate the PLL as type-I (only one integrating pole due to the DCO frequency-to-phase conversion), where it is possible to eliminate a low-pass loop filter between the phase detector and the oscillator input, resulting in a high bandwidth and fast response of the PLL. It should be noted that conventional PLLs, such as a charge-pump-based PLL, do not truly operate in the phase domain. There, the phase modeling is only a small-signal approximation under the locked condition. Their reference and feedback signals are edge based and their closest distance is measured as a proxy for the phase error. Gardner describes this as "converting the timed logic levels into analog quantities" [7]. False frequency locking is a deficiency that directly results of not truly operating in the phase domain, and it requires extra measures, such as use of a phase/frequency detector.

## IV. Reference Clock Retiming

It must be recognized that the two clock domains as described in Section III are not entirely synchronous, and it is difficult to physically compare the two digital phase values at different time instances $t_V$ and $t_R$ without having to face metastability problems. (Mathematically, $\theta_V[i]$ and $\theta_R[k]$ are discrete-time signals with incompatible sampling times and cannot be directly compared without some sort of interpolation.) Therefore, it is imperative that the digital-word phase comparison be performed in the same clock domain. This is achieved by over-sampling the FREF clock by the high-rate DCO clock, CKV, (see Fig. 2) and using the resulting CKR clock to accumulate the reference phase $\theta_R[k]$ as well as to synchronously sample the high-rate DCO phase $\theta_V[k]$, mainly to contain the high-rate transitions. Since the phase comparison is now performed synchronously at the rising edge of CKR, the (5) and (6) ought to be re-written as follows:

$$\theta_V[k] = k \tag{11}$$

$$\theta_R[k] = k \cdot N + \theta_0 + \varepsilon[k]. \tag{12}$$

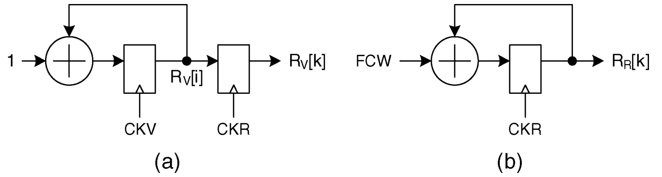

Fig. 3. Hardware implementation of (a) the variable phase $R_V[k]$ and (b) the reference phase $R_R[k]$ estimators.

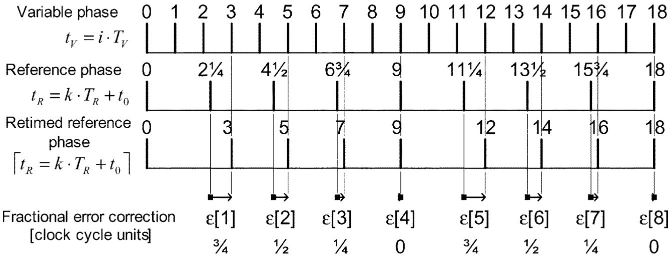

Fig. 4. Fractional-$N$ division ratio timing example with $N = 2 + (1/4)$.

The set of phase estimate [(7) and (8)] should be augmented by the sampled variable phase

$$R_V[k] = \sum_{l=0}^{i} 1 \Bigg|_{iT_V=kT_R}. \tag{13}$$

The index $k$ is the now $k$th transition of the retimed reference clock CKR, not the $k$th transition of the reference clock FREF. By constraint, it contains an integer number of CKV clock transitions. As shown in Fig. 3, $R_V[k]$ is implemented as an incrementer followed by a flip-flop register and $R_R[k]$ is implemented as an accumulator

$\varepsilon[k]$ of (12) is the CKV clock edge *quantization error*, in the range of $\varepsilon \in (0, 1)$, that could be further estimated and corrected by other means, such as the TDC-based fractional error correction circuit [3]. This operation is graphically illustrated in Fig. 4 as an example of integer-domain quantization error for a simplified case of the frequency division ratio of $N = 2 + 1/4$. Unlike $\varepsilon[k]$, which represents rounding to the next DCO edge, conventional definition of the phase error represents rounding to the closest DCO edge. An exact definition of the correction signal is not extremely important, as long as it is consistent and properly provides a negative feedback.

The reference retiming operation (shown in Fig. 2) can be recognized as a quantization in the DCO clock transitions integer domain, where each CKV clock rising edge is the next integer and each rising edge of FREF is a real-valued number. Since the system must be time-causal, quantization to the next DCO transition (next integer), rather than the closest transition (rounding-off to the closest integer), could only be realistically performed.

## V. Phase Detection

Fig. 4 suggests that the conventional definition of the phase error, expressed as a difference between the reference and variable phases, needs to be augmented here to account for the quantization correction $\varepsilon$.

$$\phi_E[k] = \theta_R[k] - \theta_V[k] + \varepsilon[k]. \tag{14}$$

Additionally, the unit of radian is not very useful here because the loop operates on the whole and fractional parts of the variable period and true unitless variables are more appropriate.

The initial temporary assumption made in Section III about the actual clock periods to be constant or time-invariant could now be relaxed at this point. Instead of producing a constant ramp of the detected phase error $\phi_E$, the phase detector will now produce an output according to the real-time clock timestamps.

The phase error can be estimated in hardware by the phase detector operation defined by

$$\widehat{\phi}_E[k] = R_R[k] - R_V[k] + \varepsilon[k]. \tag{15}$$

## VI. Modulo Arithmetic of Phase-Domain Signals

The variable and reference accumulators $R_V[i]$ and $R_R[k]$, respectively, are implemented in modulo arithmetic in order to practically limit wordlength of the arithmetic components. In this implementation, the integer part of the accumulators is $W_I = 8$ and the fractional part of the reference accumulator is $W_F = 15$. These accumulators represent the variable and reference phases, $\theta_V$ and $\theta_R$, respectively, which are linear and grow without bound with the development of time. The registers, on the other hand, cannot hold unbounded numbers, so they are restricted to a small stretch of line from zero to infinity, which repeats itself indefinitely such that any such stretch is an alias of the fundamental stretch from 0 to $2^{W_I}$.

A good example of such an approximation is modulo arithmetic. Any accumulator of length $W_I$, whose carry out bits are simply disregarded and which does not perform saturation, is a modulo-$2^{W_I}$ accumulator. In fact, the modulo arithmetic is very natural in the digital logic. Both the variable and reference modulo- accumulator. In fact, the m accumulators are not absolutely linear in the strictest sense of the word. They are, however, linear in the local sense (for a constant frequency, of course). The equations phase (7) and (8) are now rewritten to acknowledge the implicit modulo operation

$$R_V[i+1] = \mathrm{mod}\left(R_V[i] + 1, 2^{W_I}\right) \tag{16}$$

$$R_R[k+1] = \mathrm{mod}\left(R_R[k] + \mathrm{FCW}, 2^{W_I}\right). \tag{17}$$

Fig. 5 shows an example of modulo-16 operation with the frequency division ratio of $N = 10$ and exhibiting small alignment offset ($\phi_E = 3$) between the two phases. If the system is in a settled state, then both phases follow a sawtooth trajectory with the same linear speed. The variable phase $R_V[i]$ traverses all integers at a very fast pace. The reference phase $R_R[k]$, on the other hand, moves infrequently (ten times less often) but with large steps (ten times bigger). As a net effect, their traversal velocity is the same. Also shown every ten CKV clock cycles is the sampling process of $R_V[k] = R_V[i]$ during activity at $R_R[k]$. The difference between $R_R[k]$ $(3, 13, 7, 1, 11, \ldots)$ and $R_V[k]$ $(0, 10, 4, 14, 8, \ldots)$ readout sequences should always be 3, and it is so except for the forth comparison. Here, the error of $-13$ lies beyond the $(-5, 5)$ linear range and performing modulo-10 arithmetic will get it to 3, which is in line with the rest.

The modulo arithmetic on $R_V$ and $R_R$ could be visualized as two rotating vectors and the smaller angle between them constituting the phase error (Fig. 6). Both $R_V$ and $R_R$ are positive numbers and their maximum value possible without rollover

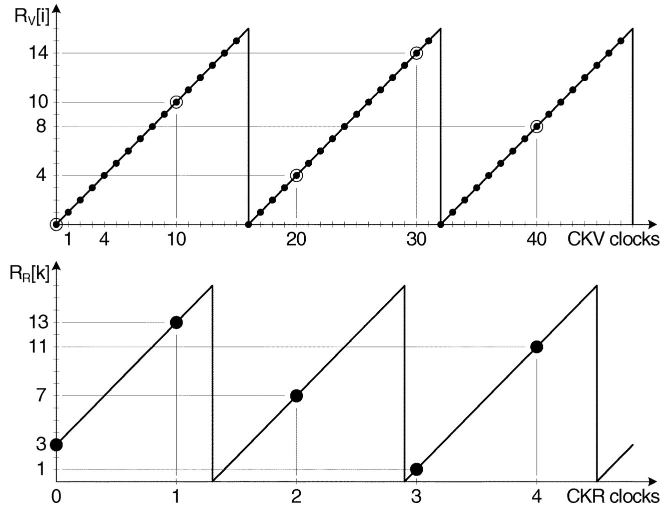

Fig. 5. Modulo arithmetic of the reference and variable phase registers with the phase offset of $\phi_E = 3$.

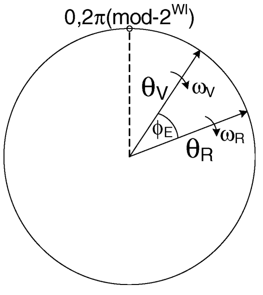

Fig. 6. Rotating vector interpretation of the reference and variable phases.

depends on the counter width or integer part of the FCW, and equals $2^{W_I}$. The phase error has the same range but is symmetric around zero, i.e., it is a two's complement number. This figure also helps to understand that the phase detector is not only the arithmetic subtractor of two numbers, but also performs a cyclic adjustment so that, under no circumstance, the larger angle between the two vectors would be decided. This could happen if, for example, the larger vector appears just before the smaller vector which is already on the other side of the "zero" radius line. Because the PD output is a $W_I$-bit constrained signed number, the conversion is always implicitly made and the output lies within $(-2^{W_I-1}, 2^{W_I-1})$. Consequently, no extra hardware is required.

## VII. Discrete-Time $z$-Domain Model

The proposed ADPLL is a discrete-time sampled system implemented with all digital components connected with all digital signals. Consequently, the $z$-domain representation is not only the most natural fit but it is also the most accurate with no necessity for those approximations that would result, for example, with an impulse response transformation due to the use of analog loop filter components [8].

Fig. 7 shows the $z$-domain model. The $\Delta\theta_V$ and $\Delta\theta_R$ are the *excess* $\theta_V$ variable (3) and $\theta_R$ reference (4) normalized transition timestamps, respectively. They are related to the conventional definition of phase, $\phi$, by $\phi_V = 2\pi \cdot \Delta\theta_V$ and $\phi_R = 2\pi \cdot \Delta\theta_R / N$. The sampling rate is the reference frequency $f_R$.

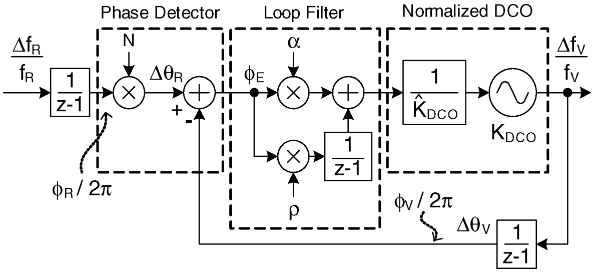

Fig. 7. $z$-domain model of the type-II ADPLL.

Two approximations are used in order to simplify the model. The first is to force the uniform sampling or PLL update rate, despite the presence of a small amount of jitter in the FREF clock. The second approximation is the linearity assumption between a frequency deviation $\Delta f$ from a center frequency $f = 1/T$ and a period deviation $\Delta T$: $\Delta f \approx f^2 \cdot \Delta T$, [1]. These two approximations are very accurate since the period deviation due to jitter and modulation is several orders of magnitude smaller than the DCO period.

The open-loop phase transfer function can be expressed as

$$H_{ol}(z) = \frac{\alpha(z-1) + \varrho}{(z-1)^2}. \tag{18}$$

For simplicity, the DCO gain estimation accuracy $r = K_{\mathrm{DCO}} / \hat{K}_{\mathrm{DCO}}$ is assumed to be unity. Otherwise, it would only result in scaling of the loop gain parameters: $\alpha \leftarrow r \cdot \alpha$ and $\rho \leftarrow r \cdot \rho$.

Since the TDC-based phase detection mechanism of the proposed architecture measures the oscillator timing excursion normalized to the DCO clock cycle [3], the frequency multiplier $N \equiv$ FCW is not part of the open-loop transfer function and, hence, does not affect the loop bandwidth. This is in contrast to conventional PLLs that use frequency division in the feedback path, but is similar, however, to the PLL architectures with frequency down-conversion (heterodyning) in the feedback path. Phase deviation of the FREF, on the other hand, needs to be multiplied by $N$ since it is measured by the same phase detection mechanism normalized to the DCO clock cycle. The same amount of timing excursion on the FREF input translates into a larger phase by a factor of $N$ when viewed by the phase detector.

The closed-loop phase transfer function is written as

$$H_{cl}(z) = \frac{\Phi_V(z)}{\Phi_R(z)} = N \frac{H_{ol}(z)}{1 + H_{ol}(z)} \tag{19}$$

where, $\Phi(z)$ is a $z$-transform of $\phi$. It could be expanded as

$$H_{cl}(z) = N \frac{\alpha(z-1) + \varrho}{(z-1)^2 + \alpha(z-1) + \varrho}. \tag{20}$$

## VIII. Linear $s$-Domain Approximation

The $z$-operator is defined as $z = e^{s/f_R}$, where $s = j2\pi f$ and $1/f_R$ is the sampling period. For small values of $f$ in comparison with the sampling rate $f_R$, we can make the following approximation:

$$z = e^s \approx 1 + s = 1 + \frac{s}{f_R} \tag{21}$$

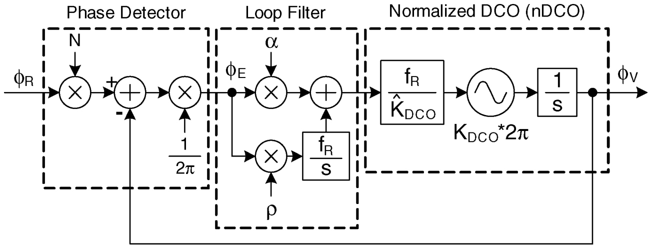

Fig. 8. Linear $s$-domain model of the type-II ADPLL.

which results in

$$s = f_R(z-1). \tag{22}$$

Using (22) to transform (18) and (20) gives

$$H_{ol}(s) = \left(\alpha + \frac{\varrho \cdot f_R}{s}\right)\frac{f_R}{s} = \frac{\varrho \cdot f_R^2}{s} \cdot \frac{1 + \frac{s}{\frac{\varrho \cdot f_R}{\alpha}}}{s} \tag{23}$$

$$H_{cl}(s) = N\frac{\alpha f_R s + \varrho f_R^2}{s^2 + \alpha f_R s + \varrho f_R^2}. \tag{24}$$

The open-loop transfer function $H_{ol}(s)$ shows two poles at origin and one complex zero at $\omega_z = j(\varrho \cdot f_R/\alpha)$. The closed-loop transfer function $H_{ol}(s)$ can be compared to the classical two-pole system transfer function

$$H_{cl}(s) = N\frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{25}$$

where, $\zeta$ is the damping factor and $\omega_n$ is the natural frequency. Fitting the two equations yields

$$\omega_n = \sqrt{\varrho} f_R \tag{26}$$

$$\zeta = \frac{\alpha f_R}{2\omega_n} = \frac{1}{2} \cdot \frac{\alpha}{\sqrt{\varrho}}. \tag{27}$$

The main motivation behind adding the integral term $\rho$ to the proportional loop gain $\alpha$ is to be able to further attenuate (up to 40 dB/dec) the lower-frequency $1/f$ noise components. The implemented ADPLL can still meet the Bluetooth specifications in the simplest type-I configuration ($\rho = 0$), but type-II setting would be required for Global System for Mobile communications (GSM).

Fig. 8 shows the $s$-domain linear model of ADPLL in the type-II setting. It is a continuous-time approximation of a discrete-time $z$-domain model and is valid as long as the frequencies of interest are much smaller (see (21); [7] reports $\leq 1/10$) than the sampling rate, which in this case equals $f_R$. There is a multiplication factor of $1/2\pi$ at the output of the phase detector. It is simply due to the fact that the digital phase error is expressed not in units of radian of the DCO clock, but in the number of its cycles.

## IX. Implementation and Results

The proposed 2.4-GHz ADPLL-based frequency synthesizer, together with the complete single-chip Bluetooth radio, are fabricated in a 0.13-$\mu$m digital CMOS process with copper interconnects, 1.5 V transistors, 0.35-$\mu$m minimum metal pitch, 2.9-nm gate oxide thickness and with no extra masks. Total continuous power consumption during TX is only 25 mA at 1.5-V supply and 2.5-dBm RF output power. The chip implementation is described in [3].

The $z$- and $s$-domain models have been validated against VHDL simulations and lab measurements [9]. The close-in synthesizer phase noise is measured $-86.2$ dBc/Hz at 10-kHz offset. The integrated rms phase noise is 0.9° (GSM spec: $\leq 5°$). Settling time is $\leq 50$ $\mu$s. The chip has passed the official Bluetooth qualification and is in production.

## X. Conclusion

We have presented mathematical description and operational details of a phase-domain ADPLL that is used as a 2.4-GHz frequency synthesizer in a commercial single-chip Bluetooth radio fabricated in a 0.13-$\mu$m CMOS process. The architecture is build from the ground up using digital techniques that exploit high speed and high density of a deep-submicrometer CMOS process while avoiding its weaker handling of voltage resolution. The conventional phase/frequency detector and charge-pump combination is replaced by a *digital-to-frequency* converter and is followed by a simple digital loop filter that controls a DCO. Modulo arithmetic is used to represent the phase-domain fixed-point signals. A novel method of retiming the reference clock allows creating a single clock domain for the synchronous operation of digital logic. Both $z$- and $s$-domain models of the ADPLL are derived, and the fit into the classical two-pole control system is given.

## References

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

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

[3] B. Staszewski, C.-M. Hung, K. Maggio, J. Wallberg, D. Leipold, and P. Balsara, "All-digital phase-domain TX frequency synthesizer for Bluetooth radios in 0.13-$\mu$m CMOS," in *Proc. IEEE Solid-State Circuits Conf.*, vol. 527, sec. 15.3, Feb. 2004, pp. 272–273.

[4] R. E. Best, *Phase Locked Loops: Design, Simulation and Applications*, 2nd ed. New York: McGraw-Hill, 1993.

[5] A. Kajiwara and M. Nakagawa, "A new PLL frequency synthesizer with high switching speed," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 407–413, Nov 1992.

[6] R. B. Staszewski, D. Leipold, and P. T. Balsara, "Just-in-time gain estimation of an RF digitally controlled oscillator for digital direct frequency modulation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 11, pp. 887–892, Nov. 2003.

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

[8] J. P. Hein and J. W. Scott, "$z$-domain model for discrete-time PLLs," *IEEE Trans. Circuits Syst.*, vol. 35, no. 11, pp. 1393–1400, Nov. 1988.

[9] R. B. Staszewski, C. Fernando, and P. T. Balsara, "Event-driven simulation and modeling of an RF oscillator," in *Proc. 2004 IEEE Int. Symp. Circuits Systems*, May 2004, pp. IV-641–IV-644.