IPSJ Transactions on System LSI Design Methodology Vol.7 2–15 (Feb. 2014)

[DOI: 10.2197/ipsjtsldm.7.2]

Invited Paper

All-Digital RF Phase-Locked Loops Exploiting Phase Prediction

Jingcheng Zhuang[1,a) Robert Bogdan Staszewski[2,b)

Received: July 30, 2013, Released: February 14, 2014

Abstract: This paper presents an all-digital phase-locked loop (ADPLL) architecture in a new light that allows it to significantly save power through complexity reduction of its phase locking and detection mechanisms. The predictive nature of the ADPLL to estimate next edge occurrence of the reference clock is exploited here to reduce the timing range and thus complexity of the fractional part of the phase detection mechanism as implemented by a time-to-digital converter (TDC) and to ease the clock retiming circuit. In addition, the integer part, which counts the DCO clock edges, can be disabled to save power once the loop has achieved lock. It can be widely used in fields of fractional-N frequency multiplication and frequency/phase modulation. The presented principles and techniques have been validated through extensive behavioral simulations as well as fabricated IC chips.

Keywords: all-digital PLL (ADPLL), digitally controlled oscillator (DCO), digital-to-time converter (DTC), phase-locked loop (PLL), phase prediction, frequency multiplication, frequency synthesis, frequency modulation, time-to-digital converter (TDC), phase modulation

1. Introduction

The past several years have seen proliferation of all-digital phase-locked loops (ADPLL) for RF and high-performance frequency synthesis due to their clear benefits of flexibility, reconfigurability, transfer function precision, settling speed, frequency modulation capability, and amenability to integration with digital baseband and application processors [1], [2]. When implemented in nanoscale CMOS, the ADPLL also exhibits advantages of better performance, lower power consumption, lower area and cost over the traditional analog-intensive charge-pump PLL [3], [4], [5].

As the ADPLLs are now employed in more and more high-volume consumer applications, there is a continuous push to provide state-of-the-art performance at ever lower cost and power consumption. This paper focuses on the implementation of the ADPLL based on true phase-domain operation, with techniques to reduce the implementation complexity and the power consumption.

The organization of this paper is as follows, Section 2 gives an overview of a digital approach to the RF frequency synthesis in the form of an all-digital PLL (ADPLL) and its phase-domain operation principle. Section 3 covers the implementation of a new generalized all-digital phase-locked loop architecture that allows it to significantly save power through complexity reduction of its phase locking and detection mechanisms. The DTC and TDC gain calibration is discussed in Section 4, followed by behavior model/simulation results in Section 5. Finally, Section 6 concludes this paper.

2. ADPLL Operation Principles

2.1 ADPLL Categories

The published ADPLLs fall into two major architectural types: the feedback-divider-based topology [6], [7], [8], [9], as shown in **Fig. 1** (a); and feedback-divider-less counter-based topology [3], [4], [5], [10], [11], [12], [13], [14], [15], [16], as shown in Fig. 1 (b). The latter type got developed first, however, the former type has shown great appeal to the established PLL community due to its topological similarity with the traditional fractional-N charge-pump PLL [17] with $\Sigma\Delta$ dithering of the modulus divider [18].

In both of these architectures, a traditional VCO got directly replaced by a digitally-controlled oscillator (DCO) for generating an output variable clock (CKV), a traditional phase/frequency detector (PFD) and a charge pump got replaced by a time-to-digital converter (TDC) for detecting phase departures of the variable clock versus the frequency reference (FREF) clock, and an analog loop RC filter got replaced with a digital loop filter for appropriately bringing the DCO into the frequency and phase lock. The conversion gains of the DCO and TDC circuits are readily estimated and compensated in the background using "free" but powerful digital logic.

What differentiates between the two ADPLL architectures is how the variable clock CKV is fed back into the TDC for the purpose of phase detection/estimation. In Fig. 1 (a), the CKV is edge divided such that its *average* frequency is ideally the same as the frequency f_R of FREF clock. The noise-shaped dithering of the modulus divider is employed to achieve an arbitrary

[1] Qualcomm Technologies, Inc., San Diego, CA, USA
[2] Delft University of Technology, Delft, The Netherlands
[a)] jingchengzhuang@ieee.org
[b)] r.b.staszewski@tudelft.nl

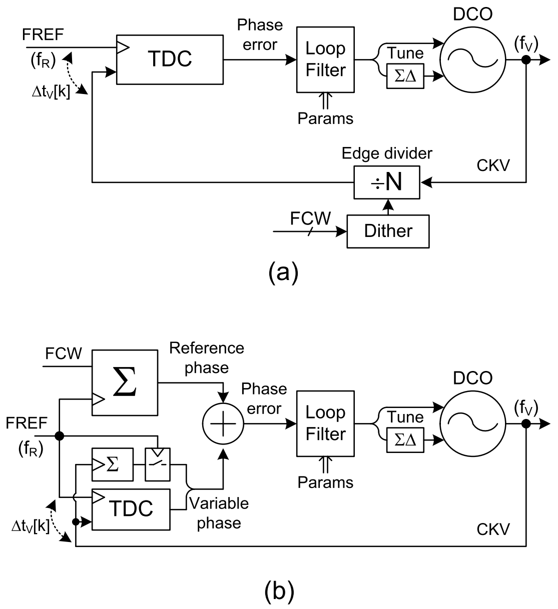


Fig. 1 ADPLL types: (a) divider-based ADPLL mimicking the conventional charge-pump PLL with ΣΔ dithering of the modulus divider; (b) divider-less counter-based ADPLL. The rest of the paper concentrates on the latter type.

CKV frequency, thus forcing the TDC range to be substantially increased[*1]. In addition, the phase error at the TDC output will exhibit high-frequency noise that needs to be attenuated by the loop filter, thus placing constraints on its filtering characteristics. Furthermore, the type-II configuration is required, otherwise the timing separation between the FREF and down-divided CKV will not be minimized, which will put enormous stress on the TDC linear range. Lastly, an additional frequency detector capability is needed during the frequency settling, otherwise the TDC would be required to cover the full range of $T_R = 1/f_R$. The last three issues are overcome at the cost of higher hardware complexity and additional constraints on the system design.

The ADPLL architecture of Fig. 1 (b) does not exhibit the recited problems. It natively handles the fractional frequency ratio, without any need for dithering. The CKV is directly connected to the TDC. As no CKV dithering is needed, the TDC covers a narrow range of the CKV period, which is much smaller than the FREF period. The TDC range is readily extended at the system level through a CKV-edge counter [19]. In this interpretation, the TDC is redefined as a timestamp-to-digital converter and now features a large dynamic range. Its output is a fixed-point number consisting of the integer count of the CKV cycles and the normalized fractional (in the units of CKV cycles) separation between the CKV and FREF edges. At the fundamental level, the ADPLL shown in Fig. 1 (b) operates in the true phase domain [20], [21] by comparing the variable phase of the multi-GHz digitally-controlled oscillator (DCO) with the reference phase of the lower-frequency (e.g., 8–40 MHz) FREF clock of high long-term precision. The comparison result is a digital phase error which, after filtering by the digital loop filter, adjusts the DCO frequency in the negative feedback manner.

The ADPLL of Fig. 1 (b) has proven its cost, power consumption and performance benefits over the traditional approaches and is currently used in worldwide production of about 33% of new mobile phones. References [3], [22], [23] and [24] describe implementations of the ADPLL-based commercial RF-SoC's for Bluetooth (130 nm CMOS), GSM (90 nm CMOS) and EDGE (65 nm CMOS) wireless standards, respectively. ADPLL implementations also include Refs. [6], [7], [8], [10], [11], [12], [13], [14] and [25].

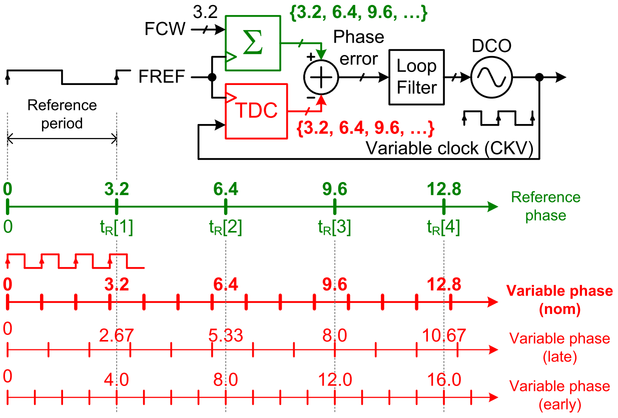


Fig. 2 Principle of the phase-domain operation of the ADPLL of Fig. 1 (b). TDC is redefined as a timestamp-to-digital converter that contains both integer and fractional parts of the variable phase.

It should be noted that there have been reported other all-digital PLL implementations, such as Refs. [26], [27], [28], [29] and [30], but they target clock generation rather than wireless RF carrier generation. The requirements of phase noise and spurious tones are much tougher in the wireless applications, hence, the designs constraints are entirely different.

2.2 ADPLL Based on Phase Domain Operation

Figure 2 explains the phase domain operation of the ADPLL (Fig. 1 (b)). The frequency reference information is wholly contained in the transition times (i.e., timestamps)[*2] of the frequency reference (FREF) clock. Of the two possible transition types, only rising clock edges are used here[*3]. Likewise, the timing information of the high-frequency variable clock (CKV) is contained in its rising edge timestamps. For the sake of illustration, the frequency command word (FCW), denoting the *expected* frequency multiplicative ratio, is 3.2. Since the oscillation time period is an inverse of the oscillating frequency, there will be 3.2 clock cycles of CKV per single cycle of FREF. Also, we assume the initial phase to be zero (i.e., FREF and CKV rising edges are aligned at time zero), although, in general, it does not need to be the case.

The phase domain operation is based on numerically calculating the phase error $\phi_E[k]$, which is a difference between the reference phase $R_R[k]$ and variable phase $R_V[k]$. The unit of the phase calculation, also called unit interval (UI), is the CKV clock pe-

[*1] The TDC linear range must be further increased by at least several CKV clock cycles depending on the type and order of the ΣΔ dithering used.

[*2] It could be beneficial to use the shape information of the reference waveform, particularly when it is of the regular sinusoidal shape as generated by the crystal oscillator (XO) but it is more complex and requires either continuous-time operation or oversampling of the continuous-time reference.

[*3] It could be beneficial to use both rising and falling edges for the phase error estimation but it is more complex and the non-50% duty cycle needs to be accounted for.

riod. Hence, the reference phase signifies the *expected* number of CKV cycles from the time zero (i.e., calculated as a summation of FCW: $R_R[k] = \sum FCW[k]$), whereas the variable phase signifies their *actual* number. In other words, the difference between the actual and ideal count of CKV cycles at each reference edge is a measure of phase *departure* or phase error, $\phi_E[k] = R_R[k] - R_V[k]$. The phase error then adjusts the DCO frequency and phase in the negative feedback manner.

A small inconsistency in the reasoning logic might possibly be noticed here. The variable clock CKV period, rather than the more stable FREF period, is the unit measure of the $R_R[k]$ and $R_V[k]$ phase quantities even though the CKV is *subject to change* due to noise and possible change in FCW. Despite this apparent paradox, the system works properly since the error correction mechanism is the *difference* between these two phase quantities. As an example, the phase error needs to go higher (i.e., DCO needs to speed up), if the variable phase gets lower (i.e., DCO gets slower) or the reference phase gets higher (i.e., more CKV cycles per FREF cycle). Assuming the FREF clock is stable, as it is supposed to be, and FCW is constant, both of these cases are equivalent to the DCO getting slower. In case the FCW increases, the DCO is *requested* to speed up.

2.3 Glossary

The following list provides a quick reference of the most common terms associated with ADPLL.

- Phase-locked loop (PLL) – Frequency synthesizer based on a negative feedback loop that generates an output "variable" signal that is related to the phase of an input "reference" signal.
- All-digital phase-locked loop (ADPLL) – A PLL consisting of key component elements with only digital inputs and outputs.
- Phase-prediction ADPLL (PP-ADPLL) – An ADPLL that predicts edge positions of the phase detector inputs in order to lower complexity of its analog circuitry.
- Frequency reference (FREF) – A signal external to the PLL that provides stable reference from which the output variable frequency and phase are derived.
- Frequency command word (FCW) – A digital signal that controls the ADPLL frequency multiplication ratio.
- Variable clock (CKV) – A clock synthesized by a PLL based on the FREF frequency and FCW.
- Reference and variable phase – Phase of FREF and CKV in the units of nominal cycles of the variable clock.
- Phase error – Difference between the reference and variable phase.
- Digitally controlled oscillator (DCO) – An oscillator in which the frequency tuning control is *fully* digital.
- DCO gain – A conversion gain of a DCO, which is actually a form of a DAC, of a frequency deviation versus input code in the units of Hz/LSB.
- Time-to-digital converter (TDC) – Phase detector that digitizes the time difference between the reference and variable clocks.
- Digital-to-time converter (DTC) – Delays edges of a clock (here: reference clock) by an amount set by its digital input.
- Loop filter – A filter connected between the phase detector and a variable oscillator to control the PLL characteristics.
- Gear shifting – Instantaneous change of the ADPLL loop bandwidth.
- Frequency modulation – Deliberate change in frequency of the synthesized signal; used to convey information.
- Digital RF – Implements the desired RF functionality using digital and digitally-intensive techniques rather than more conventional analog-intensive approaches. It exploits the following characteristics of the scaled CMOS technology: high speed and density of digital circuits, switching and matching characteristics of MOS transistors, high integration density of MOS and MOM capacitors. ADPLL is such a "digital RF" realization of a PLL.
- RF system-on-chip (RF-SoC) – Combines RF, analog and digital functionalities of an entire system in a single chip.
- RF built-in selft-test (RF-BIST) – A technique that allows an RF-SoC chip to test itself. In an ADPLL it usually involves observing the digital phase error signal.

3. ADPLL Implementation

The ADPLL block diagram of Fig. 1 (b) is now redrawn in **Fig. 3** with more implementation details. The DCO shows not a single but actually three tuning word inputs to separately control the three varactor banks: process, voltage, temperature (PVT) centering; acquisition and tracking. The PVT bank ("P") re-centers the DCO natural frequency to the middle of the selected frequency band. The acquisition bank ("A") performs channel selection by quickly settling to the neighborhood of the desired frequency. The tracking bank ("T") is the one actually used during the mission-mode transmission or reception. The ADPLL quickly transverses the P/A/T varactor banks with progressively finer frequency steps (GSM example: 4 MHz, 200 kHz and 12 kHz, respectively) while significantly narrowing down the loop bandwidth at each step. This way, the settling time can be extremely fast (i.e., several μs[*4]) and largely independent from the initial frequency difference. To maintain a certain control of the ADPLL filtering characteristics, each of the three tuning inputs has

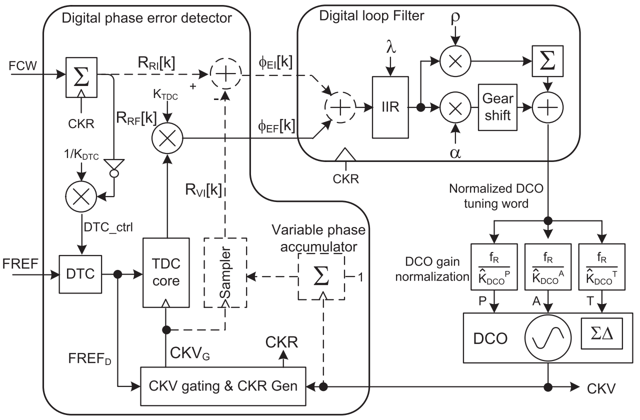


Fig. 3 Detailed block diagram of the ADPLL. The phase detector block introduces advanced phase-prediction concepts [31].

[*4] Optimized settling times of 5 μs and 3 μs are reported in Refs. [12] and [32], respectively.

its own DCO gain estimation normalizing multiplier $f_R/\widehat{K}^X_{DCO}$, where X = P, A, T. The accuracy of K^P_{DCO} and K^A_{DCO} is not very critical. For example, 10% error of their value can lead to only 10% change in the loop bandwidth and acquisition time.

The loop filter (described in more detail later in Section 3.2) consists of a $4th$-order IIR filter followed by a proportional-integral (PI) controller that includes the proportional gain factor α and integral gain factor ρ. The attenuator factor α establishes the PLL loop first-order filtering characteristic: $f_{BW} = \alpha \cdot f_R/2\pi$, where f_{BW} is a 3-dB cut-off frequency of the closed PLL loop. For example, in a Bluetooth operation, where the IIR filter is not used, the α value is changed several times during the frequency locking with an initial $\alpha = 2^{-3}$ and final $\alpha = 2^{-8}$ values resulting in f_{BW} = 259 kHz and f_{BW} = 8 kHz, respectively, for the f_R = 13 MHz reference frequency. The final value of α was chosen to be the best trade-off between the phase noise of the reference input and the DCO phase noise during the transmit (TX) and receive (RX) operations. The integral loop factor $\varrho = 2^{-18}$ is activated shortly after the loop is settled. It switches the PLL characteristic from type-I to type-II with the damping factor $\zeta = \frac{1}{2}(\alpha/\sqrt{\varrho}) \approx 1$ in order to effectively filter out the oscillator flicker noise, which tends to be quite high in scaled CMOS.

In the digital phase error detector, FCW is first accumulated to create a digital reference phase, $R_R[k]$, which is then compared with the DCO variable phase, $R_V[k]$, to obtain the digital phase error. The integer and fractional portions of the phase error detection are implemented separately using phase prediction technique to improve performance and power efficiency, which is elaborated in Section 3.1.

3.1 Digital Phase Error Detection

Digital phase error detection circuit is mainly responsible for generating the phase error (in digital format) based on the reference clock (FREF) phase, DCO clock (CKV) phase and the frequency control word (FCW). It consists of the integer part (based on a CKV cycle counter) and the fractional part (based on a TDC) operation. For implementation simplicity, generation of the re-timed reference clock (CKR) for digital operation is also included in the digital phase error detection building block.

3.1.1 Principle of Phase Prediction

To achieve closed-loop phase noise performance at a level required for most wireless and other high-performance systems, the time resolution of the TDC is usually on the order of 10–20 ps [19]. To cover at least one CKV period with such a uniform time resolution, an inverter-based TDC operates by delaying the CKV edges through a chain of inverters (with a total delay larger than T_V) and sampling the inverter chain outputs at each rising edge of FREF. The sampled digital state contains the information of the CKV-FREF timing separation in the units of the inverter delay. Such a TDC usually requires a fairly significant number (e.g., tens to hundreds) of delay cells and flip-flops to achieve the desired timing resolution and range. In addition, delaying CKV edges through the inverter chain is power hungry because all delay cells toggle at the high CKV frequency.

To alleviate the above issue, a phase prediction method, whose principle is illustrated in **Fig. 4**, is introduced into the ADPLL [15], [16], [31], [39]. The top two lines show timestamps of the CKV and FREF clock rising edges, respectively, for an example FCW = $2\frac{1}{4}$. The units are one CKV period or 360° of the generated clock. Due to the fractional FCW part being nonzero, the timing deviation between the FREF edge and the next CKV edge shows a periodical pattern of 0, $\frac{3}{4}$, $\frac{1}{2}$, $\frac{1}{4}$, 0, etc, with the repetition period of 4. In the conventional ADPLL of Fig. 1 (b), the TDC needs to cover the worst-case of the FREF-CKV timing deviation, which is one CKV period. Now, with the phase prediction technique, the FREF edge gets delayed such that it is always aligned with the next CKV edge. This way, the TDC would need to cover a much smaller range of only a few quantization levels (i.e., one or two orders of magnitude of an improvement) just to account for phase noise (i.e., jitter) and errors in the delay control.

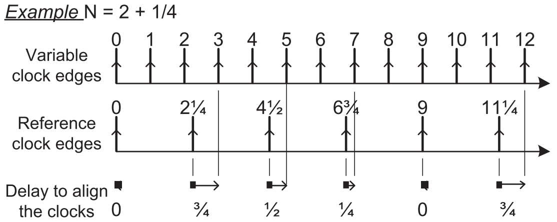


Fig. 4 Principle of the phase prediction in which the next timing separation (i.e., phase) between the FREF and next CKV edges are predicted based on FCW and the current phase.

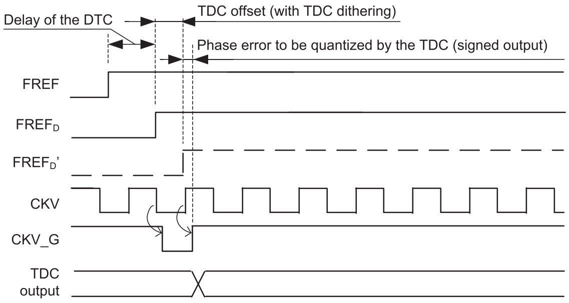

Delay of the DTC
TDC offset (with TDC dithering)
Phase error to be quantized by the TDC (signed output)
FREF
FREF_D
FREF_D'
CKV
CKV_G
TDC output

Fig. 5 Timing diagram of the phase prediction technique.

As the timing diagram in **Fig. 5** shows, the reference clock (FREF) is passed through a digital-to-time converter (DTC) to generate a delayed version of the reference clock (FREF$_D$) (see the digital phase error detection part of Fig. 3). The amount of delay is based on $R_{RF}[k]$(the fractional part of the reference phase, $R_R[k]$). The TDC compares the edge of FREF$_D$ with the edge of CKV_G, which is a gated CKV clock (using FREF$_D$, see Section 3.1.3 for details) and has the same average frequency as the reference clock for low-power operation. In the figure, a virtual signal FREF$_D$' is shown and the delay between FREF$_D$' and FREF$_D$ represents the TDC offset. The value of the TDC offset does not affect the phase noise performance as long as it is constant. When the PLL is in the phase-locked condition, FREF$_D$' is dynamically aligned with the next "safe" edge of CKV or CKV$_G$ (so-called phase prediction), and a narrow TDC can be employed to quantize the dynamic phase error ($\phi_{EF}[k]$), which is thus added[*5] into $\phi_{EI}[k]$ (the integer part of the phase error ϕ_E) to obtain the total phase error so that the overall time res-

[*5] Depending on the design target, the adder may be simply replaced with a MUX, please refer to Section 3.2 for more details.

olution is equal to the resolution of the narrow TDC. Because of the reduced required operating range of the TDC, various high-resolution TDC circuit topologies may be employed without burning significant amount of power. With the fractional part enabled, the integer phase error $\phi_{EI}[k]$ is always zero as long as the loop stays in the locked condition and all circuitry related to the integer part (i.e., dashed lines in Fig. 3) may be disabled to significantly reduce the power consumption.

Another benefit of the phase prediction method is that the signal FREF$_D$ is also used to gate the CKV clock before it enters the narrow TDC so that the TDC only operates on the edges required for time quantization to minimize its power consumption (Fig. 5). In addition, the fact that the signal FREF$_D$ is approximately synchronized with the CKV edges eases the CKR generation circuit and reduces the power consumption. A detailed discussion on the CKV clock gating and CKR generation circuit is given in Section 3.1.3.

In summary, the main advantages of the phase prediction technique are:

(1) The integer part of the phase error ($\phi_{EI}[k]$) keeps zero in the normal operation so it can be turned off after the PLL frequency acquisition to reduce the power consumption;

(2) It significantly reduces the required operating range of the TDC, and makes it feasible to employ various high-resolution TDCs to achieve better noise performance;

(3) Both DTC and TDC operate at reference rate (the minimum possible rate for the phase error detection) without the need for additional power management circuitry [14], [19], thus resulting in significant power savings;

(4) The fractional part of the phase error, $\phi_{EF}[k]$, may be measured (and normalized based on TDC resolution) directly from the narrow TDC;

(5) Because the signal FREF$_D$ is aligned with CKV, the complexity and the power consumption of CKR generation circuit and CKV gating clock circuit is significantly reduced.

3.1.2 Phase Prediction Block

Upon reaching the phased-locked condition, the T_V-normalized time difference between the FREF and CKV edges is stochastically equal [*6] to R_{RF} as:

$$\mathcal{E}\left\{\frac{t_R - t_V}{T_V} - R_{RF}\right\} = 0 \tag{1}$$

where $\mathcal{E}$ denotes the statistical expectation operator, t_R and t_V represent timestamps of the FREF and CKV edges, and T_V is the CKV period. To find out the desired delay of the DTC, Eq. 1 is rewritten as:

$$[t_R + T_V \times (1 - R_{RF})] - t_V = T_V \tag{2}$$

The non-zero value on the right-hand side of Eq. 2 signifies that the FREF edge is delayed by $T_V \times (1 - R_{RF})$ such that it aligns with the *next* edge of CKV, rather than the current edge, which is required for time causality. With the normalized DTC gain K_{DTC} defined as $\Delta t_{DTC}/T_V$, where Δt_{DTC} is the LSB delay step size of the DTC, the desired DTC control can be expressed as:

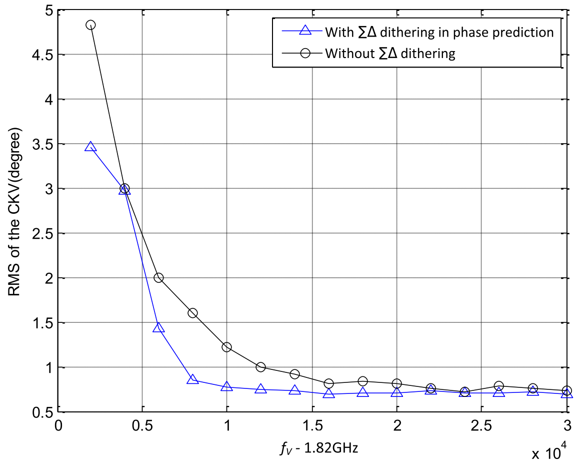


Fig. 6 Impact of the ΣΔ dithering in the phase prediction.

$$DTC_{ctrl,fp} = \frac{(1 - R_{RF})}{K_{DTC}} \tag{3}$$

The integer part of the calculated fixed-point value DTC$_{ctrl,fp}$ is used as the DTC control code, labeled as DTC_ctrl, as shown in Fig. 3, in which the operation of $(1 - R_{RF})$ is achieved by bit inversion. If desired, the performance/quantization resolution may be improved (not shown in Fig. 3) if the fractional part of DTC$_{ctrl,fp}$ is either:

(1) dithered into the integer part using a digital ΣΔ modulator when the TDC resolution Δt_{TDC} is similar or coarser than the DTC resolution Δt_{DTC} (this also includes the case of a 1-bit TDC); or

(2) converted to residue and added into the fine TDC output if the TDC resolution is much finer than the DTC resolution.

In option 1, dithering the fractional part of the desired DTC delay value of DTC$_{ctrl,fp}$ into the integer control code will reduce the effect of the DTC quantization noise through shaping the quantization noise into high frequency so that it can be better attenuated by the digital loop filter. This has an especially significant effect when the DTC quantization noise is ill-behaved (i.e., not white in the frequency domain), which appears when the FREF-CKV timing separation does not significantly vary over time.

The above observation is quantified as an example in **Fig. 6**, which plots root-mean-square (RMS) phase noise values obtained through behavioral closed-loop simulations, with a first-order ΣΔ dithering of the DTC (i.e., option 1) turned either on and off, versus the PLL locking frequency separation from the integer-N channel of 1,820 MHz (FCW = 1,820 MHz / 26 MHz = 70). The frequency channels are spaced by 200 kHz. The ΣΔ dithering of DTC reduces the rms phase noise for near-integer frequency channels while it has negligible impact when fractional part of FCW is far from zero (or one[*7]), in which case other noise sources predominate. Similar conclusion was reached in Ref. [33], which investigated the adverse effects of near-integer-N PLL operation and dithering as a means to mitigate them. Note that the DTC-TDC pair could be viewed as a complex TDC comprising two stages: coarse (i.e., full-range DTC) and fine (i.e., narrow-range TDC). Alternatively, the first stage could be made even coarse by

[*6] Here only the variation is considered because the constant difference does not affect the loop operation.

[*7] 1.0 aliases to the fractional value of 0.0 and the next integer.

selecting the closest phase of the quadrature (i.e., 90° separated) CKV clock [15].

In the case where the TDC resolution is much finer than the DTC resolution, the phase prediction residue, $0.5 - (DTC_{ctrl,fp} - DTC_{ctrl})$, is added into the fine digital TDC output so that the DTC resolution does not limit the overall time resolution.

3.1.3 CKV Gating and CKR Generation Circuit

As shown in Fig. 3, the reference phase accumulator, edge predictor, digital loop filter and so on, are clocked by the retimed reference clock, CKR, which is running on average at the reference frequency f_R and is edge-synchronized with the CKV edge. To generate such a clock, a straight-forward method would be to sample the reference clock with the CKV clock. There are two apparent concerns with that approach:

(1) Because CKV clock edges are not synchronized with the reference edges, the direct sampling may result in a metastable output when the sampling happens precisely at the edge of the reference clock;

(2) Although the CKR output only toggles once after each reference edge, the sampler is clocked at the high CKV frequency, thus burning a lot of unnecessary power.

Concern 1 can be solved by sampling the reference clock with two parallel sampling flip-flops triggered by rising and falling edges of the CKV, and then choosing the output further away from metastability based on an arbitration signal from the TDC [5]. Despite ensuring the metastability-free operation, it increases the hardware complexity and power consumption. For concern 2, the unnecessary power consumption could be reduced by disabling the CKR generation circuit between the edges of the reference clock with a timer circuit. However, this may also consume additional power. In this work, the CKR generation is automatically gated using the delayed reference clock and it runs at the reference clock rate without additional timer circuitry or exhibiting any metastability issues.

Figure 7 (a) shows the CKR generation circuit together with the TDC clock gating circuit. In this figure, I_5 and I_6 are asynchronously resettable flip-flops. As the timing diagram in Fig. 7 (b) shows, before the rising edge of $FREF_D$, CKV_{EN} stays low and disables the OR gate (I_1) to keep CKV_1 high, regardless of the CKV level. On the rising edge of $FREF_D$, CKV_{EN} becomes high, allowing the CKV edge pass through I_1 so the first CKV rising edge after the $FREF_D$ results in a rising edge at CKV_1 and further triggers I_5 and creates a rising edge on CKR_2. The rising edge of CKR_2 resets I_6 and CKV_{EN} goes back to low to disable the high-activity path from CKV to CKV_1 in order to minimize the power consumption. The falling edge of $FREF_D$, resets I_5 to create a falling edge at CKR_2 in preparation for the next rising edge of $FREF_D$.

The signal CKR_2 is re-timed twice by CKVD8 (CKV divided by 8) and buffered to produce CKR, as shown in Fig. 7 (a). The delay between CKR_2 and CKR is more than $8 \times T_V$, enough time for TDC and variable phase accumulator to determine the fractional part of the phase error ($\phi_{EF}[k]$), the integer part of variable phase ($R_{VI}[k]$) and the phase error ($\phi_E[k]$). Equivalently, CKR, being a re-timed and buffered version of CKR_2, is generated by sampling $FREF_D$ with CKV rising edges without any flip-flop running at CKV frequency. In fact, only one input of the logical gate (I_1) is toggled at CKV rate while the remaining circuits run at the reference frequency (one rising edge and one falling edge in each reference period). Because $FREF_D$ is synchronized with CKV edges using the phase prediction technique, the metastability problem of flip-flops is thus avoided in this CKR generation circuit.

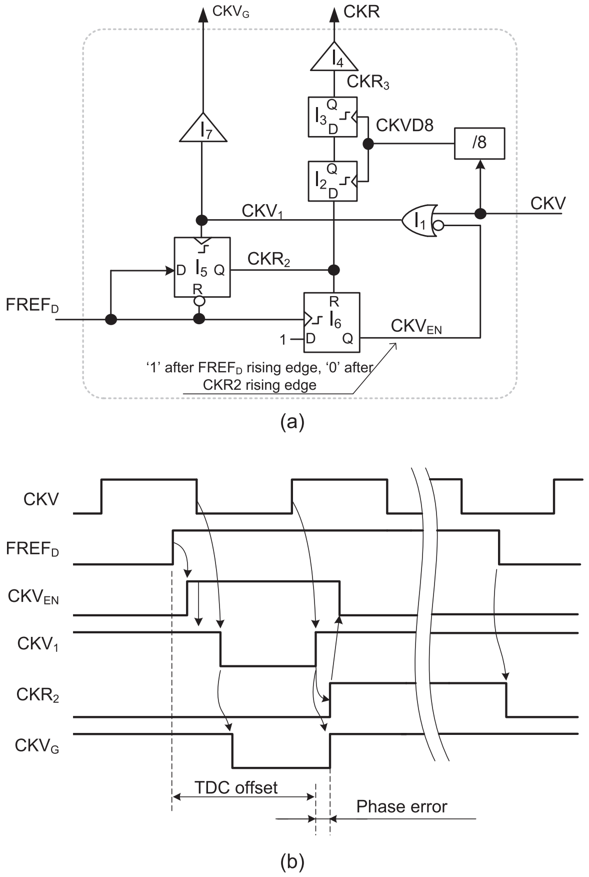


Fig. 7 CKR generation and clock gating circuit.

To reiterate, the circuit shown in Fig. 7 is responsible for gating the input signal for the narrow TDC to minimize the power consumption. After the rising edge of $FREF_D$, the CKV rising edges appear at CKV_1, which is then buffered to generate CKV_G. Consequently, both inputs of the TDC, i.e., CKV_G and $FREF_D$, are running at the reference rate, thus significantly reducing the TDC power consumption. In this CKR generation and clock gating circuit, the number of toggling edges is minimized and thus the power consumption is maximally reduced.

The phase prediction scheme and clock gating circuit operates based on an assumption that the timing error seen by the narrow TDC is bounded within a narrow fraction of the CKV period when the PLL is in the phase-locked condition, which certainly holds true for a great majority of practical applications. However, for some applications exhibiting high amount of noise, the oscillator and the reference experience excessive instantaneous timing error (more than $T_V/2$) between $FREF_D$ and CKV even with the phase prediction method described above. In this case, the clock gating circuit shown in Fig. 7 may potentially pick up wrong CKV edge. To avoid this problem, the signal CKV_1 may be generated from CKV using a pulse swallower (controlled by the carry bit of

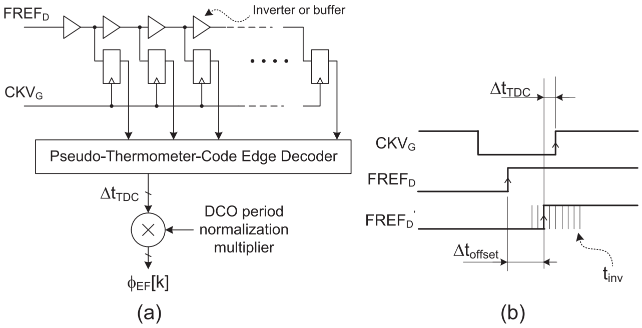


Fig. 8 Time-to-digital converter (TDC) core: (a) structure; (b) quantization of the timing difference between the FREF$_D$ and CKV$_G$ edges.

the fractional part digital reference phase accumulator) followed by an integer frequency divider. Further details on this approach are omitted as out of scope of this paper.

3.1.4 Time-to-Digital Converter (TDC)

The TDC, as shown in **Fig. 8**, generates the fractional part of the phase error (ϕ_{EF}) by quantizing the time difference between FREF$_D$ and CKV$_G$ edges. Unlike previously reported TDCs, which generate the variable phase (fractional part) or timestamps of the FREF edges in the units of the DCO clock period T_V [19], the TDC shown in Fig. 3 actually quantizes the timing error to generate fractional part of the digital phase error (without additional subtraction with the reference phase, R_R). Thus, the interpreted TDC output is signed, as opposed to the unsigned encoding in the conventional ADPLL designs.

As shown in Fig. 8 (a), the delayed reference clock (FREF$_D$) gets delayed by the string of inverters or buffers, whose outputs are sampled with the rising edge of the gated CKV clock (CKV$_G$). The obtained TDC core output forms a pseudo-thermometer code (as illustrated in Fig. 8 (b)), which is then converted to binary. The value expresses the FREF$_D$-CKV$_G$ separation in the units of the inverter delay t_{inv} (i.e., being an equivalent to the TDC resolution Δt_{res} in this architecture). Due to the phase prediction nature of the ADPLL, the number of TDC inverters can be set to cover far less than one T_V.

Since t_{inv} is subject to process, voltage and temperature (PVT) shifts, the T_V-normalization is usually required as shown in Fig. 8 (a) and Fig. 3. Because the TDC output represents the phase error (instead of variable phase [19]), the DCO period normalization may be implemented in the digital loop filter together with the loop gain multipliers to reduce the hardware complexity. Such a normalization needs an estimated $K_{TDC} = t_{inv}/T_V$ and the estimation method is discussed in Section 4.

3.2 Digital Loop Filter (DLF)

Since the conventional phase/frequency detector and charge pump, which encode the phase error by the width of the train of pulses at FREF rate, are replaced by the TDC, the phase-domain operation does not fundamentally generate any reference spurs thus allowing for the *digital* loop filter to be set at an optimal performance point between the reference phase noise and the oscillator phase noise. Consequently, the ADPLL for Bluetooth [3] is merely designed to provide only the first or second order filtering, in contrast to the third-order filtering for the traditional PLL's. The cellular systems, however, require better filtering, and a 6^{th}-order filtering is used for GSM to sharply attenuate phase noise at the protected 400 kHz frequency offset [4]. Such sharp filtering would not be possible in a controlled manner with the traditional PLL's.

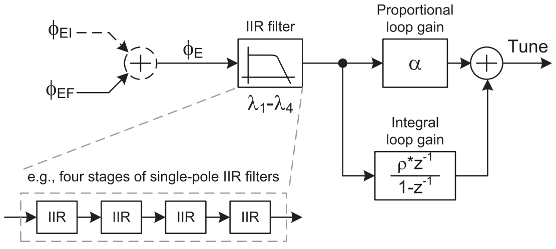


Fig. 9 Digital loop filter details.

The loop filter configuration is shown in **Fig. 9**. It consists of a phase error combiner, a cascade of four single-pole IIR filters with coefficients $\lambda_1 ... \lambda_4$, and a proportional path with loop gain coefficient α, and an integral path with loop gain coefficient ρ.

The phase error combiner determines the total phase error (ϕ_E) based on the fractional (ϕ_{EF}) and integer (ϕ_{EI}) parts of the phase error. Although this can be simply achieved using a binary adder (as shown in Fig. 9), the phase error combiner may be implemented as a multiplexer to select ϕ_{EI} during the loop acquisition process and to select ϕ_{EF} once the loop is in the locked condition.

Each single-pole IIR filter satisfies the following equation:

$$y_i[k] = (1 - \lambda_i) \cdot y_i[k-1] + \lambda_i \cdot x_i[k] \quad (4)$$

where $x_i[k]$ and $y_i[k]$ are the inputs and outputs, respectively, of each stage i with coefficient λ_i.

The proportional and integral paths are configured in parallel to create a so-called proportional-integral (PI) control structure. The PI structure is proceeded by the IIR filter, whose purpose is to further improve the transition band rejection of the ADPLL filtering characteristics.

Because of the fully-digital nature of the phase error correction, sophisticated control algorithms through a dynamic change of the loop filter parameters (refer back to Fig. 3) could be employed, which would not have been feasible with conventional architectures:

(1) Adaptable and reconfigurable characteristic of the ADPLL loop depending on the communication channel conditions or quality of the DCO and FREF clocks.

(2) Dynamic gear shifting of the ADPLL bandwidth to speed up the frequency settling [34] and to respond to unexpected and expected disturbances in the SoC, such as ramping up the power amplifier and digital baseband (DBB), keyboard or display activities.

(3) Freezing the ADPLL loop for a short interval in order not to respond at all to the expected disturbances but rather "coast" over them. This is easily accomplished through differentiating of the phase detector by moving the FREF accumulator to after the phase detector: $(\sum FCW[k] - R_V[k]) \rightarrow \sum(FCW[k] - (R_V[k] - R_V[k-1]))$. The ADPLL transfer function is the same in both cases, except for the possibly different integration constant, which has effect only on the mean value of the phase error.

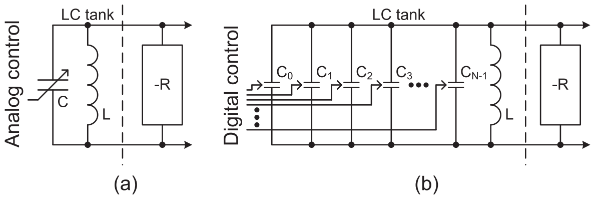


Fig. 10 LC tank based-oscillators: (a) conventional with analog control; (b) with all-digital control. The negative resistance $-R$ perpetuates the lossy LC tank resonance.

(4) Dynamic change of the ADPLL loop characteristics, such as dynamically switching from type-I to type-II loop after the settling is complete. To avoid the zero-forcing behavior of the type-II loop after the switchover from type-I, a residue method can be applied, in which the error minus the sampled value is integrated rather than the error itself.

3.3 Digitally Controlled Oscillator (DCO)

At the heart of the ADPLL lies the DCO. It is based on an LC-tank with a negative resistance to perpetuate the oscillation – just like the traditional voltage-controlled oscillator (VCO) on **Fig. 10** (a). However, there is a significant difference in one of the components: instead of continuously-tuned varactor (variable capacitor), the DCO now uses a large number of binary-controlled varactors (see Fig. 10 (b)), as first proposed in Ref. [35]. Each varactor can be placed in either high or low capacitative state. The composite varactor performs digital-to-capacitance conversion (DCC). Since the varactors, i.e., the DCO input, are digitally controlled, and since the output clock at multi-GHz frequencies is still almost of an acceptable digital waveform shape (the rise and fall times could be as fast as 30 ps), the loop around the DCO, which adjusts its phase and frequency, could now be *fully digital*, as first proposed in Ref. [36].

The finest varactor step size made possible by the fine lithography is on the order of 40 aF (i.e., 40E-18 F), which corresponds to 12 kHz frequency step size at the 2 GHz DCO output. This is equivalent to the fine control of about 250 electrons leaving and entering the LC-tank. Unfortunately, this fine control is not sufficient for any commercial wireless standard, so dithering is used that improves the *time-averaged* capacitative resolution. A typical realization, which uses a second-order MASH ΣΔ modulator [35] running at 2 GHz/8 clock rate with 8 fractional input bits, will produce the sufficiently-fine open-loop resolution of 12 kHz/256 = 45 Hz, which is now equivalent to about one electron.

Figure 11 shows a simplified schematic of the DCO core that operates in the 3.2–4.0 GHz range. The high-band and low-band cellular frequencies are obtained by means of rail-to-rail dynamic edge dividers. The tuning control is split into several banks of varying degree of frequency step size and range: coarse d^P for process, voltage and temperature (PVT) calibration; medium d^A for channel acquisition; and fine d^T for tracking of the oscillator drift. The d^P frequency range is the largest since it has to cover all the frequency bands and margin for the oscillator variability. The capacitor banks are built using MIM and MOS varactors. The oscillator phase noise is controlled by the dissipated current, which is established by the 7-bit "bias" control. In order to avoid real biasing current sources, the M_0 transistor array operates in linear (i.e., triode) region instead of in saturation. The current is set through automatic calibration at a minimum value at which the oscillator still produces the acceptable RF phase noise. It should be noted that other oscillator structures have been recently reported, such as a class-F oscillator [37], that can obtain the same low phase noise but at a much lower current consumption.

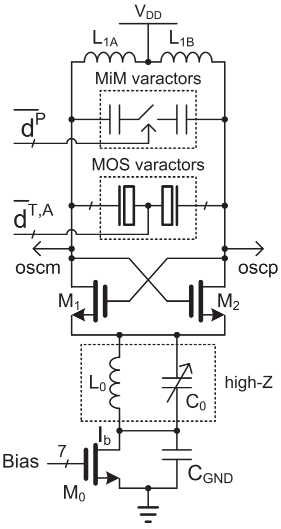


Fig. 11 Oscillator core and the varactor state driver array (GSM version example).

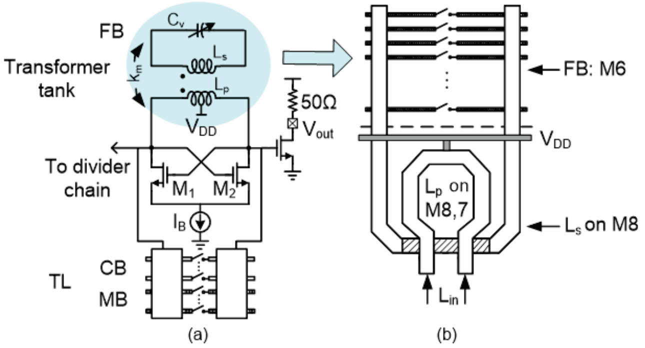


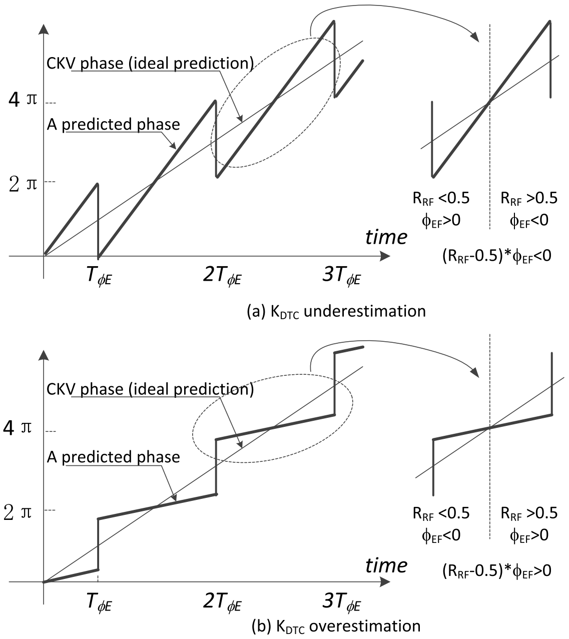
Fig. 12 60 GHz DCO: (a) schematic; (b) top layout view of the fine-tuning bank.

The fully digital manner of frequency control can be extended to a mm-wave, as demonstrated in Ref. [38] for a 60 GHz DCO (see **Fig. 12**) that is used in an ADPLL for FMCW radar [32].

4. K_{DTC} and K_{TDC} Estimation

To determine the DTC control code, Eq. (3) requires that the DTC gain, K_{DTC}, be known (either directly or indirectly). An error in the K_{DTC} estimation can lead to a phase noise degradation but it will not affect the frequency locking. The principle of K_{DTC} estimation [39] is described in this section.

To simplify the design, the DTC usually employs similar delay cell elements as those in the TDC so that the value of K_{TDC} equals to K_{DTC}, or there is a constant ratio between K_{TDC} and K_{DTC} over the process, voltage and temperature (PVT). In addition, in the phase-prediction based digital phase detection block, K_{TDC} inaccuracy is equivalent to the inaccuracy of the the loop gain and does not significantly degrade the closed loop output spectrum. Consequently, the following section focuses on the estimation of



(a) K_{DTC} underestimation

(b) K_{DTC} overestimation

Fig. 13 Operating principle of the K_{DTC} estimation.

K_{DTC}.

As mentioned above, the ADPLL is able to achieve its lock even in face of inaccurate K_{DTC}. Once it settles, the CKV output phase tracks the average predicted phase due to the heavy low-pass filtering effects of the loop. The phase error due to the inaccurate phase prediction, as a result of the inaccurate K_{DTC}, is a sawtooth waveform with a repetition frequency $f_{\phi E}$ related to fractional part of FCW (FCW$_F$) as:

$$f_{\phi E} = f_R \times min(FCW_F, 1 - FCW_F) \qquad (5)$$

where f_R is the frequency of the reference clock FREF. $f_{\phi E}$ approaches zero when FCW$_F$ is near zero, while it reaches its maximum of $f_R/2$ when FCW$_F$ is 0.5.

Figure 13 shows the operational principle of the K_{DTC} estimation, in which $T_{\phi E}$ is the period of the phase error (i.e., $T_{\phi E} = 1/f_{\phi E}$). The diagram suggests that the K_{DTC} estimation/calibration be done by detecting the estimation error and iteratively updating the estimated K_{DTC}.

Figure 13 (a) shows the case when K_{DTC} is underestimated. The fractional part of the phase error ϕ_{EF} is positive when R_{RF} is below 0.5 and ϕ_{EF} is negative when R_{RF} is larger than 0.5. ϕ_{EF} has opposite polarities in the case when K_{DTC} is overestimated, as shown in Fig. 13 (b). Note that this zero-mean of averaged ϕ_{EF} is the above-mentioned natural property of a type-II PLL. Consequently, by monitoring the polarity of the phase error ϕ_{EF} and correlating it with the known value of the reference phase R_{RF}, the estimated K_{DTC} can be iteratively updated. As a result, the actual value of $\widehat{K}_{DTC}$ ($\widehat{x}$ is an estimation of a random variable x) will be forced to gradually approach the ideal or expected value of K_{DTC}. Generally, more accurate $\widehat{K}_{DTC}$ results in less induced phase error and better closed-loop phase noise performance, which will be confirmed through simulations in Section 5.

Figure 14 shows a block diagram of the K_{DTC} estimation method. The fractional reference phase (R_{RF}) is subtracted by 0.5 and multiplied by the sign of ϕ_{EF} to generate the estimation error, which is further filtered by an IIR filter and integrated to obtain the estimated K_{DTC} or $\widehat{K}_{DTC}$. The IIR filter is of the first order and has the following equation: $IIR_{out}[k] = IIR_{out}[k-1] * (1 - 2^{-a}) + [(R_{RF}[k] - 0.5) * sign(\phi_{EF}[k])] * 2^{-b}$, where k is the discrete-time index, b is the input scaling factor and a is the feedback scaling factor. The IIR filter output is then multiplied by the step size μ of the iterative adaptation algorithm. The K_{DTC} estimation block is triggered by the CKR running at the reference rate and may be disabled once the K_{DTC} estimation is done, or kept running to track the K_{DTC} variation due to temperature or voltage changes. The effectiveness of the K_{DTC} estimation is confirmed by simulation in Section 5.

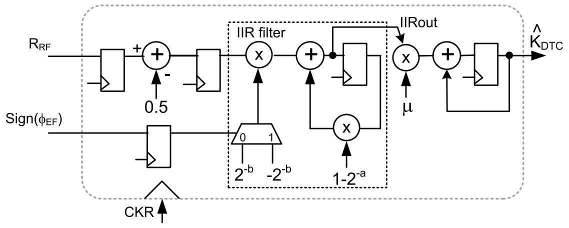


Fig. 14 Block diagram of the K_{DTC} estimation.

Although the above K_{DTC} estimation method is specifically designed for the ADPLL based phase-prediction technique, it could also be applied to conventional ADPLL architectures that require an accurate K_{TDC} estimation. In Refs. [19] and [13], the K_{TDC} estimation is a result of a non-iterative calculation that involves a fixed-point divider, which makes it more complex. In fact, this apparent complexity has led to develop a new class of a higher-complexity TDC that does not require normalization [40]. However, the iterative method described in this paper could reduce that digital complexity while keeping the TDC simple. In this case, the input of the K_{TDC} error detection circuit is the difference between the normalized (using an estimated K_{TDC}) phase and the reference phase coming from the FCW accumulator. A related adaptive estimation method of a DCO gain within a conventional ADPLL was described in Ref. [41].

5. Behavioral Simulation

The ADPLL architecture described above is modeled and simulated in Matlab using time-domain event-driven principles [42], [43]. For simplicity, the PLL is configured as a type-II second-order loop with proportional and integration paths (without the IIR filter of Fig. 9) in the digital loop filter. Depending on various design targets, higher-order loop filters (see Section 3.2) may also be employed.

5.1 Frequency and Phase Acquisition

With the reference frequency of f_R = 26 MHz and FCW = 69.2308 (f_V = 1.8 GHz), the locking process of the ADPLL is simulated with example results shown in **Fig. 15**. In this simulation, the delay resolution of DTC (Δt_{DTC}) is 15 ps. While the total phase error (ϕ_E, as shown in Fig. 15 (b)) approaches zero, the instantaneous DCO frequency (Fig. 15 (a)) reaches its target value of 1.8 GHz from the initial frequency of about 2 GHz. In this simulation, a TDC with 6 steps (±0.5, ±1.5, ±2.5), having an identical time resolution as the DTC, is modeled. In the first 40 μs,

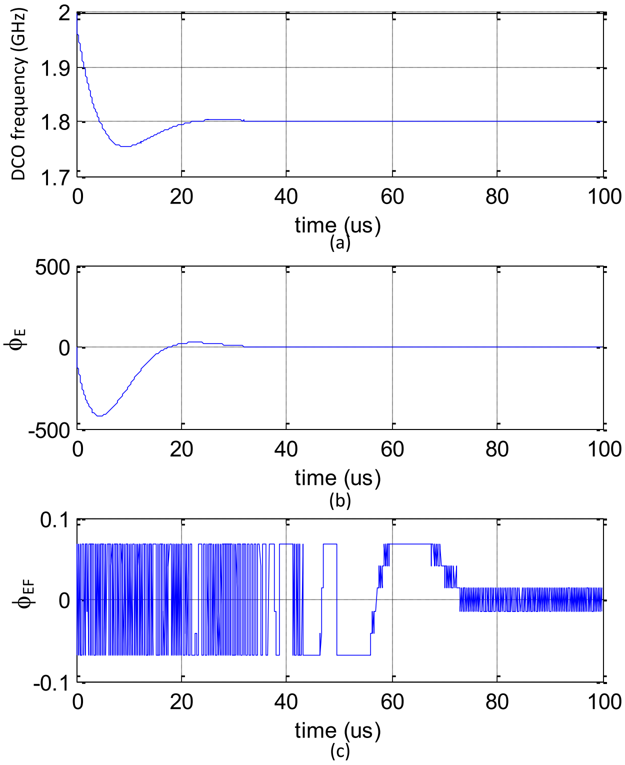


Fig. 15 ADPLL frequency/phase acquisition.

because of the limited operational range of the narrow TDC, ϕ_{EF} is bounded within ±2.5×K_{TDC}. Fortunately, during the frequency acquisition period, the total phase error is dominated by the integer phase path, hence the limited range of the TDC has negligible effect on PLL frequency acquisition process.

From 40 μs to 70 μs, the total PLL phase error is gradually dominated by the fractional part of the phase error (ϕ_{EF}), while the integer part becomes zero. After 70 μs, the integer phase error stays zero and the PLL is solely governed by the fractional phase path.

5.2 Phase Noise Performance

The phase noise performance of the ADPLL is simulated in time-domain by introducing realistic phase noise sources into the DCO and the reference clock. The loop bandwidth is set to approximately 100 kHz and all other parameters stay the same as before. The PLL runs for 4 ms and the time-domain edge jitter is converted to frequency-domain phase noise (plotted in **Fig. 16**) through spectral estimation routines with 30 kHz resolution bandwidth. In this simulation, the K_{DTC} calibration is disabled and the correct K_{DTC} value is used. As expected, the PLL efficiently suppresses the DCO phase noise within the loop bandwidth while the out-of-band phase noise is dominated by the DCO phase noise.

In the locked condition, inaccurate K_{DTC} may enlarge the phase error at the input of the narrow TDC because the phase prediction is based on an incorrect K_{DTC}. Such enlarged phase error samples are then quantized by the narrow TDC and filtered by the digital loop filter before modulating the DCO frequency and phase. Consequently, the actual impact of the K_{DTC} inaccuracy depends on the frequency content of the phase error and the characteristics of the digital loop filter. As illustrated in Fig. 13 and Eq. (5), the fundamental frequency of the phase error (resulted from K_{DTC} estimation error) reaches its maximum value of $f_R/2$ when $FCW_F = 0.5$. Due to the low-pass loop characteristics, the K_{DTC} error may have more impact on the closed-loop phase noise performance when FCW_F is near zero or one (i.e., near-integer channels). To confirm the analysis, different K_{DTC} errors are introduced in the phase prediction block and the phase noise performance for different K_{DTC} errors are simulated as shown in **Fig. 17**, in which the FCW is constant at 69.2308 (1.8 GHz/26 MHz). One can observe increasing the phase noise degradation when K_{DTC} inaccuracy increases, especially for the in-band phase noise where the digital loop filter does not attenuate.

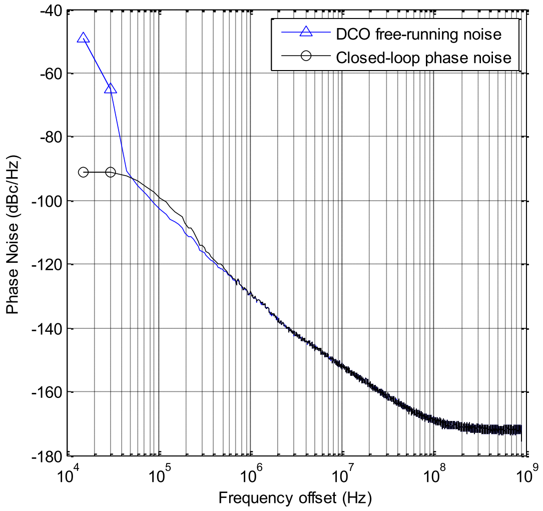


Fig. 16 PLL phase noise simulation result.

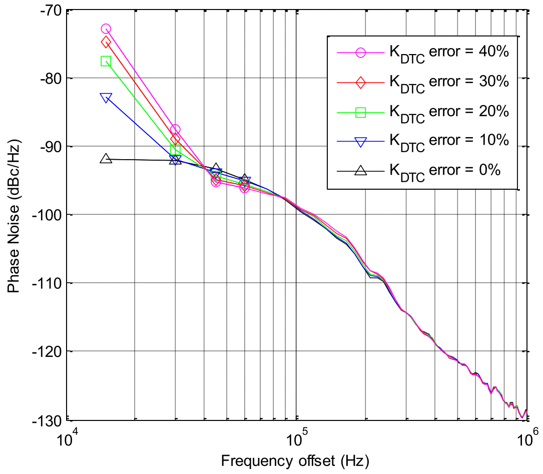


Fig. 17 PLL phase noise performance versus uncorrected K_{DTC} estimation error.

Because the fundamental frequency of the phase error caused by the inaccurate K_{DTC} is lower when the FCW is near integer values, the impact of the K_{DTC} inaccuracy is more significant when the PLL operates at near integer FCWs. **Figure 18** shows the RMS phase error (in the units of degree) of the PP-ADPLL output for different FCWs near the integer of 69. The FCW step size in this simulation corresponds to an output frequency step of 100 kHz. In this simulation, a noiseless DCO, a clean reference clock and a TDC with a resolution of 1 ps are employed to better

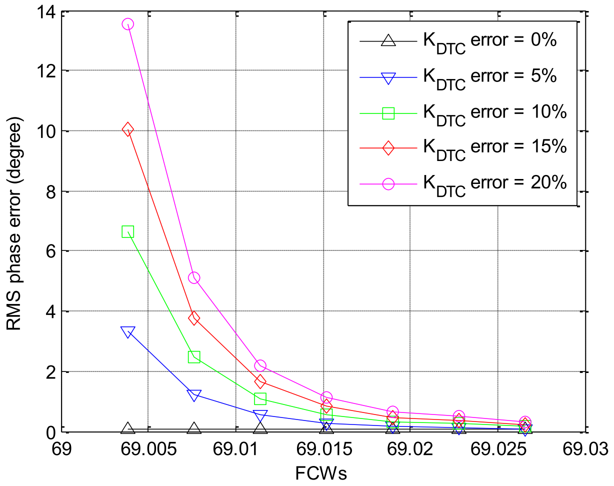


Fig. 18 RMS phase error versus FCW for different K_{DTC} errors.

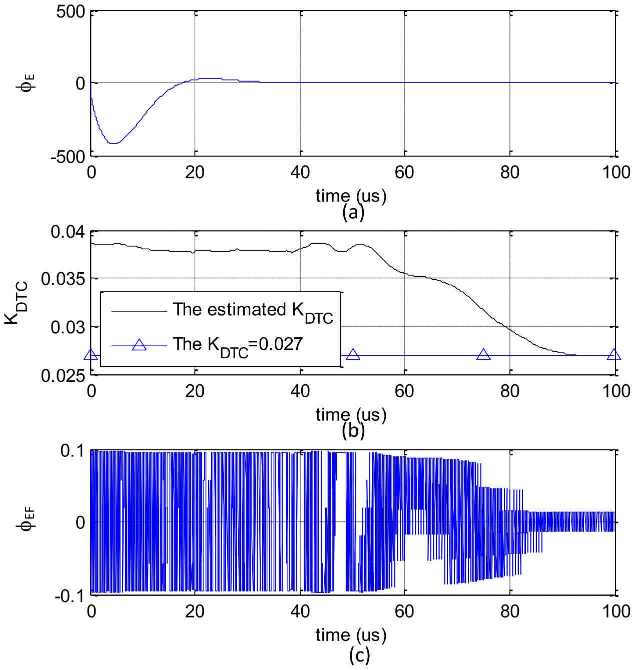


Fig. 19 ADPLL settling and K_{DTC} estimation in face of initial K_{DTC} error of 40%.

observe the impact of the K_{DTC} error. One can see the sensitivity of the RMS phase error on the K_{DTC} error increases dramatically when FCWs approach an integer.

5.3 K_{DTC} Estimation

The phase prediction requires the knowledge of K_{DTC} (see Eq. (3)). With Δt_{DTC} of 15 ps and the DCO frequency of 1.8 GHz, the K_{DTC} value is 0.027, which is then used in the phase prediction block for the simulation results shown in Fig. 15. In reality, the exact K_{DTC} is unknown because Δt_{DTC} is realized with delay elements whose delay may depend on the process, temperature and voltage (PVT) variations as well as cell-to-cell mismatches. Thus, the methodology described in Section 4 may be used to estimate K_{DTC}. With the same loop configuration as above, the PLL locking process is simulated again with the iterative K_{DTC} estimation enabled. **Figure 19** shows the simulation result. The initial value of the K_{DTC} is intentionally set to about 40% higher than the ideal value to observe the loop behavior and the efficiency of the K_{DTC} calibration method.

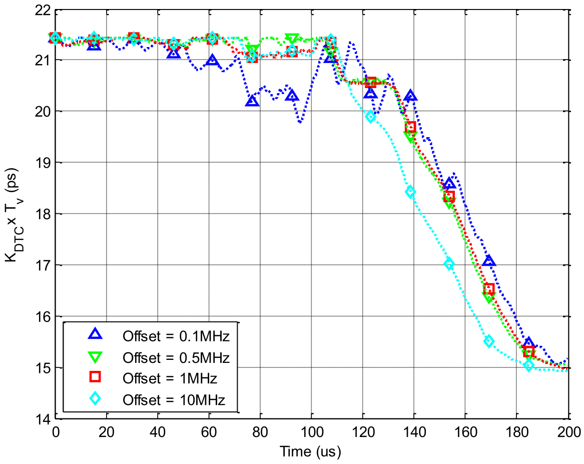


Fig. 20 K_{DTC} estimation, expressed as the estimated inverter delay in ps units, for near-integer channels in face of initial K_{DTC} error of 40%. The actual settling target is 15 ps.

Even if the K_{DTC} is off by 40% initially, the total phase error plot (Fig. 19 (a)) is dominated by the integer phase error and appears similar to the one shown in Fig. 15 (b), in which the accurate K_{DTC} value is used. However, after the integer phase error reaches zero and the loop is in the locked condition, the fractional phase error (Fig. 19 (c)) may be out of the operation range of the narrow TDC if the K_{DTC} is not accurate, as during the time interval from 20 us to 80 us in Fig. 19. While the estimated K_{DTC} (Fig. 19 (b)) approaches its actual value, the fractional phase error is lowered and finally settles within a narrow dynamic range inside the coverage of the TDC.

Figure 20 shows the K_{DTC} estimation process for different offset frequencies from the integer channel, i.e., FCW=69+Offset/f_R, with the same loop configuration as above. Because the actual K_{DTC} is different for different operational frequencies, the value of estimated Δt_{DTC}, which is the product of the estimated K_{DTC} and T_V, is plotted and compared with its actual value of 15 ps. The result shows that the K_{DTC} method works well even for near-integer channels, and the estimation error is below 1% for all cases. The impact of the estimation error is negligible based on Fig. 17 and Fig. 18. One can see that the convergence of the K_{DTC} estimation is slightly slower for near-integer channels (i.e., Offset=0.1 MHz). This is because the fundamental frequency $f_{\phi E}$ of the sawtooth waveform in the phase error (shown in Fig. 13) is lower and easier to pass through the low-pass filtering of the phase locked loop, resulting in less error energy for the K_{DTC} estimation loop.

5.4 Two-point Phase/Frequency Modulation

Similar to other existing ADPLLs, the ADPLL described in this paper is capable of the two-point phase/frequency modulation. While the frequency modulation code is added to the DCO control word after the gain normalization, the corresponding phase modulation code (only fractional part is used since, after the PLL is locked, the integer part is disabled) is added to the input of the phase prediction block (R_{RF}) so that the modulation does not introduce additional phase error at the TDC out-

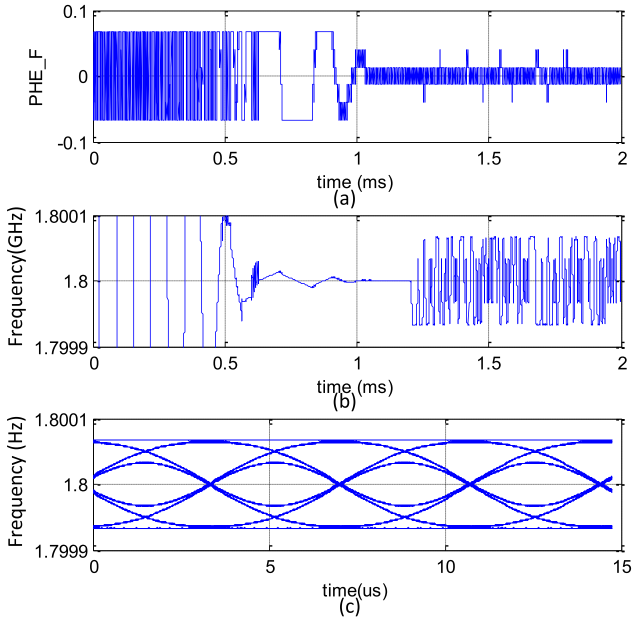


Fig. 21 Two-point frequency modulation of the ADPLL.

put. To confirm this operation, a Gaussian Minimum Shift Keying (GMSK) frequency modulation driven by a pseudo-random binary sequence (PRBS) is introduced to the ADPLL, and the instantaneous DCO frequency and its eye diagram are plotted in **Fig. 21**. The ADPLL starts with the frequency acquisition and the fractional part of phase error (Fig. 21 (a)) approaches zero after 1 ms when the DCO frequency (Fig. 21 (b)) approaches its target of 1.8 GHz. The integer path of the PLL is disabled after 1 ms. At 1.2 ms, the GMSK modulation starts and there is no significant change on the phase error plot because the two-point modulation automatically cancels the phase error caused by the modulation. The eye diagram of the instantaneous frequency in the time range from 1.3 ms to 2 ms is shown in Fig. 21 (c), which confirms the proper operation of the GMSK modulation.

6. Conclusions

In this paper, we have described the traditional all-digital phase-locked loop (ADPLL), which is now being used in a significant share of commercial mobile phones. We then pointed out inefficiencies in the digital phase error detection mechanism while introducing a phase-prediction all-digital PLL (PP-ADPLL) architecture. The new architecture uses a phase prediction technique in order to delay the reference clock edge by a predicted amount such that it is always maximally aligned with the variable clock edge. This way, the time-to-digital converter (TDC) can be of narrow range just to cover the reference and oscillator jitter and account for the delay control errors. The conventional TDC, which is typically the most power-hungry block in the ADPLL after the DCO, is thus advantageously split into a digital-to-time converter (DTC) and a narrow-range TDC. The DTC handles the predictive part, while the TDC covers the stochastic part of the phase detection operation. An added benefit of the reference clock delay is that its timing relationship with the variable clock is now precisely known, which allows to be retimed by the variable clock without the conventional issues of metastability. The PP-ADPLL also exploits inefficiencies in the conventional phase error processing mechanisms and partitions it in a such a way that the integer part can be disabled after the lock has been achieved, thus further reducing the power consumption. The advantages of the latest PP-ADPLL have been demonstrated by behavioral model simulations and also verified in silicon [15], [16].

References

[1] Staszewski, R.B. and Balsara, P.T.: *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*, New Jersey, John Wiley & Sons, Inc. (Sept. 2006).

[2] Staszewski, R.B.: State-of-the-art and future directions of high-performance all-digital frequency synthesis in nanometer CMOS, *IEEE Trans. Circuits and Systems I*, Vol.58, No.7, pp.1497–1510 (July 2011).

[3] Staszewski, R.B., Muhammad, K., Leipold, D., et al.: All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS, *IEEE Journal of Solid-State Circuits*, Vol.39, No.12, pp.2278–2291 (Dec. 2004).

[4] Staszewski, R.B., Wallberg, J., Rezeq, S., et al.: All-digital PLL and transmitter for mobile phones, *IEEE J. Solid-State Circuits*, Vol.40, No.12, pp.2469–2482 (Dec. 2005).

[5] Staszewski, R.B., Waheed, K., Dulger, F. and Eliezer, O.: Spur-free multirate all-digital PLL for mobile phones in 65 nm CMOS, *IEEE Journal of Solid-State Circuits*, Vol.46, No.12, pp.2904–2919 (Dec. 2011).

[6] Zhuang, J., Du, Q. and Kwasniewski, T.: A 4 GHz low complexity ADPLL-based frequency synthesizer in 90 nm CMOS, *IEEE Custom Integrated Circuits Conf.* (*CICC*), pp.543–546 (Sept. 2007).

[7] Hsu, C.-M., Strayer, M.Z. and Perrott, M.H.: A low-noise, wide-BW 3.6 GHz digital $\Sigma\Delta$ fractional-N synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation, *IEEE Solid-State Circuits Conf.*, pp.340–341 (Feb. 2008).

[8] Chang, H.-H., Wang, P.-Y., Zhan, J.-H. and Hsieh, B.-Y.: A Fractional Spur-Free ADPLL with Loop-Gain Calibration and Phase-Noise Cancellation for GSM/GPRS/EDGE, *Proc. IEEE Solid-State Circuits Conf.*, sec.10.1, pp.200–2001 (Feb. 2008).

[9] Wang, P.-Y., Zhan, J.-H., Chang, H.-H. and Hsieh, B.-Y.: An analog enhanced all digtial RF fractional-N pll with self-calibrated capability, *IEEE Custom Integrated Circuits Conference, 2008* (*CICC*), pp.749–752 (2008).

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

[11] Lee, M., Heidari, M.E. and Abidi, A.A.: A low-noise wideband digital phase-locked loop based on a coarse-fine time-to-digital converter with subpicosecond resolution, *VLSI Symp. Circuits*, pp.2808–2816 (Oct. 2009).

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

[13] Xu, L., Lindfors, S., Stadius, K. and Ryynanen, J.: A 2.4-GHz low-power all-digital phase-locked loop, *IEEE J. Solid-State Circuits*, Vol.45, No.8, pp.1513–1521 (Aug. 2010).

[14] Tokairin, T., Okada, M., Kitsunezuka, M., et al.: A 2.1-to-2.8-GHz low-phase-noise all-digital frequency synthesizer with a time-windowed time-to-digital converter, *IEEE Journal of Solid-State Circuits*, Vol.45, No.12, pp.2582–2590 (Dec. 2010).

[15] Lai, J.-W., Wang, C.-H., Kao, K., Lin, A., Cho, Y.-H., Cho, L., Hung, M.-H., Shih, X.-Y., Lin, C.-M., Yan, S.-H., Chung, Y.-H., Liang, P., Deng, G.-K., Li, H.-S., Chien, G. and Staszewski, R.B.: A 0.27 mm^2 13.5 dBm 2.4 GHz all-digital polar transmitter using 34%-efficiency class-D DPA in 40 nm CMOS, *Proc. IEEE Solid-State Circuits Conf.* (*ISSCC*), pp.342–343 (Feb. 2013).

[16] Chillara, V.K., Liu, Y.-H., Wang, B., Ba, A., Vidojkovic, M., Philips, K., de Groot, H. and Staszewski, R.B.: An 860 μW 2.1-to-2.7 GHz all-digital PLL-based frequency modulator with a DTC-assisted snapshot TDC for WPAN (Bluetooth Smart and ZigBee) applications, *Proc. IEEE Solid-State Circuits Conf.* (*ISSCC*), sec.9.8 (Feb. 2014). (accepted)

[17] Gardner, F.M.: Charge-pump phase-locked loops, *IEEE Trans. Communications*, Vol.COMM-28, pp.1849–1858 (Nov. 1980).

[18] Riley, T., Copeland, M. and Kwasniewski, T.: Delta-sigma modulation in fractional-N frequency synthesis, *IEEE Journal of Solid-State Circuits*, Vol.28, No.5, pp. 553–559 (May 1993).

[19] Staszewski, R.B., Vemulapalli, S., Vallur, P., Wallberg, J. and Balsara, P.T.: 1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS, *IEEE Trans. Circuits and Systems II*, Vol.53, No.3, pp.220–224 (Mar. 2006).

[20] Kajiwara, A. and Nakagawa, M.: A new PLL frequency synthesizer with high switching speed, *IEEE Trans. Vehicular Technology*, Vol.41, No.4, pp.407–413 (Nov. 1992).

[21] Staszewski, R.B. and Balsara, P.T.: Phase-domain all-digital phase-locked loop, *IEEE Trans. Circuits and Systems II*, Vol.52, No.3, pp.159–163 (Mar. 2005).

[22] Staszewski, R.B., Leipold, D., Eliezer, O., Entezari, M., Muhammad, K., Bashir, I., Hung, C.-M., Wallberg, J., Staszewski, R., Cruise, P., Rezeq, S., Vemulapalli, S., Waheed, K., Barton, N., Lee, M.-C., Fernando, C., Maggio, K., Jung, T., Elahi, I., Larson, S., Murphy, T., Feygin, G., Deng, I., Mayhugh, T., Ho, Y.-C., Low, K.-M., Lin, C., Jaehnig, J., Kerr, J., Mehta, J., Glock, S., Almholt, T. and Bhatara, S.: A 24 mm^2 quad-band single-chip GSM radio with transmitter calibration in 90 nm digital CMOS, *Proc. IEEE Solid-State Circuits Conf.*, pp.208–209, 607 (Feb. 2008).

[23] Mehta, J., Staszewski, R.B., Eliezer, O., Rezeq, S., Waheed, K., Entezari, M., Feygin, G., Vemulapalli, S., Zoicas, V., Hung, C.-M., Barton, N., Bashir, I., Maggio, K., Frechette, M., Lee, M.-C., Walberg, J., Cruise, P. and Yanduru, N.: A 0.8 mm^2 all-digital SAW-less polar transmitter in 65 nm EDGE SoC, *Proc. IEEE Solid-State Circuits Conf.*, pp.58–59 (Feb. 2010).

[24] Staszewski, R.B., Waheed, K., Vemulapalli, S., Dulger, F., Walberg, J., Hung, C.-M. and Eliezer, O.: Spur-free all-digital PLL in 65 nm for mobile phones, *Proc. IEEE Solid-State Circuits Conf.*, pp.52–53 (Feb. 2011).

[25] Tonietto, R., Zuffetti, E. and Castello, R.: A 2 MHz bandwidth low noise RF all digital PLL with 12 ps resolution time-to-digital converter, *European Solid-State Circuits Conf.* (*ESSCIRC*), pp.150–153 (Sept. 2006).

[26] Mair, H. and Xiu, L.: An architecture of high-performance frequency and phase synthesis, *IEEE J. Solid-State Circuits*, Vol.35, pp.835–846 (June 2000).

[27] Chang, H.-H., Lee, S.-M., Chou, C.-W., Chang, Y.-T. and Cheng, Y.-L.: A 1.6-880 MHz synthesizable ADPLL in 0.13 um CMOS, *IEEE International Symposium on VLSI Design, Automation and Test* (*VLSI-DAT*), pp.9-12 (Apr. 2008).

[28] Tierno, J.A., Rylyakov, A.V. and Friedman, D.J.: A wide power supply range, wide tuning range, all static CMOS all digital PLL in 65 nm SOI, *IEEE J. Solid-State Circuits*, vol.43, No.1, pp.42–51 (Jan. 2008).

[29] Sai, B., Reddy, P., Krishnaprasad, N., Moorthi, S., Raja, J. and Perinbam, P.: An all digital phase locked loop for ultra fast locking, *Proc. Natl. Conf. Emerging Trends in Engineering and Technology* (Apr. 2008).

[30] Yu, G., Wang, Y., Yang, H. and Wang, H.: Fast-locking all digital phase-locked loop with digitally controlled oscillator tuning word estimating and presetting, *Circuits, Devices & Systems, IET*, Vol.4, No.3, pp.207–217 (May 2010).

[31] Zhuang, J. and Staszewski, R.B.: A low-power all-digital PLL architecture based on phase prediction, *Proc. 19th IEEE International Conference on Electronics, Circuits, and Systems* (*ICECS'12*), pp.797–800 (Dec. 2012).

[32] Wu, W., Bai, X., Staszewski, R.B. and Long, J.R.: A 56.4-63.4 GHz spurious-free all-digital fractional-N PLL in 65 nm CMOS, *Proc. IEEE Solid-State Circuits Conf.* (*ISSCC*), sec.20.4, pp.352–353 (Feb. 2013).

[33] Waheed, K., Staszewski, R.B., Dulger, F., Ullah, M.S. and Vamvakos, S.D.: Spurious-free time-to-digital conversion in an ADPLL using short dithering sequences, *IEEE Trans. Circuits and Systems I*, Vol.58, No.9, pp.2051–2060 (Sept. 2011).

[34] Staszewski, R.B. and Balsara, P.T.: All-digital PLL with ultra fast settling, *IEEE Trans. Circuits and Systems II*, Vol.54, No.2, pp.181–185 (Feb. 2007).

[35] Staszewski, R.B., Hung, C.-M., Leipold, D. and Balsara, P.T.: A first multigigahertz digitally controlled oscillator for wireless applications, *IEEE Trans. Microwave Theory and Techniques*, Vol.51, No.11, pp.2154–2164 (Nov. 2003).

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

[37] Babaie, M. and Staszewski, R.B.: A class-F CMOS oscillator, *IEEE Journal of Solid-State Circuits* (*JSSC*), Vol.48, No.12, pp.3120–3133 (Dec. 2013).

[38] Wu, W., Long, J.R. and Staszewski, R.B.: High-resolution millimeter-wave digitally controlled oscillators with reconfigurable passive resonators, *IEEE Journal of Solid-State Circuits* (*JSSC*), Vol.48, No.11, pp.2785–2794 (Nov. 2013).

[39] Zhuang, J.-C. and Staszewski, R.B.: Gain estimation of a digital-to-time converter for phase-prediction all-digital PLL, *Proc. IEEE 21th European Conference on Circuit Theory and Design* (*ECCTD'13*) (Sept. 2013).

[40] Opteynde, F.: A 40 nm CMOS all-digital fractional-N synthesizer without requiring calibration, *Proc. IEEE Solid-State Circuits Conf.*, sec.20.3, pp.346–347 (Feb. 2012).

[41] Staszewski, R.B., Wallberg, J., Hung, C.-M., Feygin, G., Entezari, M. and Leipold, D.: LMS-based calibration of an RF digitally-controlled oscillator for mobile phones, *IEEE Trans. Circuits and Systems II*, Vol.53, No.3, pp.225–229 (Mar. 2006).

[42] Staszewski, R.B., Fernando, C. and Balsara, P.T.: Event-driven simulation and modeling of phase noise of an RF oscillator, *IEEE Trans. Circuits and Systems I*, Vol.52, No.4, pp.723–733 (Apr. 2005).

[43] Zhuang, J., Du, Q. and Kwasniewski, T.: Event-driven modeling and simulation of an digital PLL, *Proc. Behavioral Modeling and Simulation Conf.* (*BMAS*), pp.67–72 (2006).

Jingcheng Zhuang received his M.A.Sc. and Ph.D. degrees in Electronics from Carleton University in 2003 and 2007 respectively. From 2001 to 2005, he worked on various challenging industrial and academic R&D projects, in the fields of DLL/PLL-based frequency synthesis, clock and data recovery and channel equalization. From 2005 to 2006, he was with Altera Corp., working on the system and circuit design of PLL/DLL-based frequency synthesizers and oversampling clock data recovery circuits. From 2006 to 2009, he was with Texas Instruments Inc., Dallas, TX, USA, and responsible for architecting, implementing and validating single-chip radios in nanoscale CMOS processes, with the focus of the all-digital PLL and RF transmitter system design. He was with Advanced Micro Devices Inc. from 2009 to 2011, working on the system and circuit design of high-speed transceiver design, and he is currently with RFIC group of Qualcomm Technologies Inc. He holds more than 40 international publications and US patents, and his research interests include deep-submicron CMOS RF architectures and circuits, analog and digital PLL-based frequency synthesis, high-speed transceivers, clock and data recovery, channel equalization, etc.

Robert Bogdan Staszewski received his B.S.E.E. (*summa cum laude*), M.S.E.E. and Ph.D. degrees from the University of Texas at Dallas in 1991, 1992 and 2002, respectively. From 1991 to 1995 he was with Alcatel Network Systems in Richardson, TX, USA, working on Sonet cross-connect systems for fiber optics communications. He joined Texas Instruments in Dallas, TX, USA, in 1995 where he was elected Distinguished Member of Technical Staff (2% of the technical population). Between 1995 and 1999, he had been engaged in advanced CMOS read channel development for hard disk drives. In 1999 he co-started a Digital RF Processor (DRP) group within Texas Instruments with a mission to invent new digitally-intensive approaches to traditional RF functions for integrated radios in deep-submicron CMOS processes. Dr. Staszewski has served as a CTO of the DRP group between 2007 and 2009. Since July 2009 he is Professor at Delft University of Technology in the Netherlands. He has authored and co-authored one book, two book chapters, 160 journal and conference publications, and holds 110 issued US patents. His research interests include nanoscale CMOS architectures and circuits for frequency synthesizers, transmitters and receivers. He is an IEEE Fellow and recipient of IEEE Circuits and Systems Industrial Pioneer Award.

(Invited by Editor-in-Chief: *Hiroyuki Tomiyama*)