

# A Fractional- $N$ Bang-Bang PLL Based on Type-II Gear Shifting and Adaptive Frequency Switching Achieving 68.6 fs-rms-Total-Integrated-Jitter and 1.56 $\mu$ s-Locking-Time

Simone M. Dartizio<sup>1</sup>, *Graduate Student Member, IEEE*,  
Francesco Buccoleri<sup>1</sup>, *Graduate Student Member, IEEE*, Francesco Tesolin<sup>1</sup>, *Graduate Student Member, IEEE*,  
Luca Avallone<sup>1</sup>, *Graduate Student Member, IEEE*, Alessio Santiccioli<sup>1</sup>, *Member, IEEE*,  
Agata Iesurum<sup>1</sup>, *Graduate Student Member, IEEE*, Giovanni Steffan, *Member, IEEE*, Dmytro Cherniak,  
Luca Bertulesi<sup>1</sup>, *Member, IEEE*, Andrea Bevilacqua<sup>1</sup>, *Senior Member, IEEE*, Carlo Samori<sup>1</sup>, *Fellow, IEEE*,  
Andrea L. Lacaita<sup>1</sup>, *Fellow, IEEE*, and Salvatore Levantino<sup>1</sup>, *Senior Member, IEEE*

**Abstract**—This work presents a fast-locking and low-jitter fractional- $N$  bang-bang phase-locked loop (BBPLL). To break the trade-off between jitter and locking time which is typical of BBPLLs, two novel techniques are introduced. A gear-shift technique, denoted as type-II gear-shift, avoids limit cycles in the phase-locked loop (PLL) frequency transient and optimizes the locking time of the main PLL loop. The adaptive frequency switching (AFS) technique reduces the PLL frequency error upon channel switching exploiting the already existing hardware. The prototype, implemented in a 28-nm CMOS process, has an active area of 0.23 mm<sup>2</sup> and achieves a locking time always below 1.56  $\mu$ s (within 80 ppm accuracy) for frequency jumps up to 1.5 GHz over the 8.5–10 GHz tuning range. The measured rms jitter (integrated from 1 kHz to 100 MHz) is 48.6 fs for integer- $N$  channels and 68.6 fs for near-integer fractional- $N$  channels, with a worst case fractional spur of  $-58.2$  dBc. The power consumption is 20 mW, leading to a jitter-power figure of merit of  $-253.2$  and  $-250.3$  dB for integer- $N$  and fractional- $N$  channels, respectively.

**Index Terms**—Bang-bang phase-locked loop (BBPLL), fast-locking, frequency switching, gear-shifting, low-jitter.

## I. INTRODUCTION

MULTIPLE-output arrays, carrier aggregation, and high-order modulation schemes are being implemented

Manuscript received 15 May 2022; revised 1 August 2022 and 3 September 2022; accepted 12 September 2022. This article was approved by Associate Editor Wanghua Wu. This work was supported by Infineon Technologies, Villach. (*Corresponding author: Simone M. Dartizio.*)

Simone M. Dartizio, Francesco Buccoleri, Francesco Tesolin, Luca Bertulesi, Carlo Samori, Andrea L. Lacaita, and Salvatore Levantino are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mail: simonemattia.dartizio@polimi.it).

Luca Avallone, Giovanni Steffan, and Dmytro Cherniak are with Infineon Technologies AG, 9500 Villach, Austria.

Alessio Santiccioli was with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy. He is now with Qualcomm, San Diego, CA 92121 USA.

Agata Iesurum and Andrea Bevilacqua are with the Dipartimento di Ingegneria dell'Informazione, University of Padua, 35131 Padua, Italy.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSSC.2022.3206955>.

Digital Object Identifier 10.1109/JSSC.2022.3206955

in wireless transceivers to support the increasing demand of higher data rates. In this frame, ultralow jitter local oscillators (LOs) are essential to meet the bit error rate requirements. For example, the 5G new-radio at the upper millimeter-wave (MMW) frequency band needs an integrated jitter less than 90 fs [1]. Such ultralow jitter LOs can be implemented with analog phase-locked loops (PLLs) [1], [2], [3], [4], [5], [6]. However, the adoption of a digital PLL (DPLL) is more attractive for its smaller footprint fully exploiting the scaling of advanced CMOS technologies [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. Among DPLLs, the bang-bang PLL (BBPLL) is even more attractive since the use of a single bit quantizer, denoted as bang-bang phase-detector (BBPD), saves additional area and power consumption while providing ultralow jitter performance [7], [8], [9], [10], [11], [12], [13], [14], [15]. Unfortunately, the narrow linear range of the BBPD, caused by its single bit output, slows down the PLL transient, thus preventing the adoption of BBPLLs where frequency agility is a key requirement in addition to jitter. Fig. 1 schematically shows a typical BBPLL frequency transient, highlighting the main limitations of this scheme. Due to the saturation of the BBPD output, the digitally controlled oscillator (DCO) frequency,  $F_{\text{out}}$ , is initially updated at a limited rate and even when the frequency error is small enough to make the phase detector (PD) output switching its sign again, the nonlinearity of the BBPD causes long residual overshoots lasting until the PLL is eventually locked. The settling transient of the BBPD scheme can be speeded-up by increasing the loop gain and the corresponding PLL loop bandwidth. Unfortunately, since the PLL integrated jitter is also a function of the loop bandwidth, this option increases the PLL jitter, which is usually not acceptable. To overcome these limitations, a high-resolution and wide-range time-to-digital converter (TDC) could be adopted [18], [19], [21], but this comes at the cost of increased power consumption and area occupation.

To break the trade-off between jitter and locking time two main strategies are reported in literature and summarized in

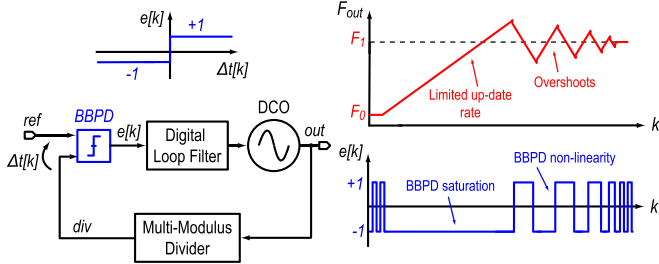


Fig. 1. Frequency transient of a BBPLL upon application of a frequency jump.

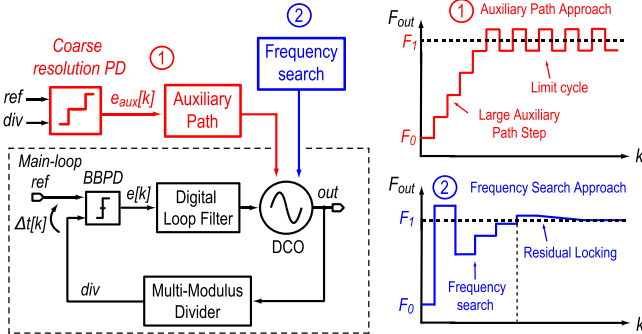


Fig. 2. Fast-locking approaches adopted in BBPLLs: 1-implementation of an auxiliary path with coarse resolution PDs and a larger bandwidth, and 2-implementation of a frequency search algorithm to quickly set the DCO frequency close to its final steady state value.

Fig. 2. The first approach is based on the use of auxiliary PDs with a coarse resolution, which allow to monitor the phase error over an extended range but with reduced hardware complexity and power dissipation with respect to a full high-resolution TDC. The phase information is exploited to increase the loop bandwidth during the locking transient. When locking is approached, the auxiliary PDs are disabled thus recovering the bandwidth for optimal jitter and a low power dissipation at steady state. Solutions in [7], [11], [13], [20], [21], [22], and [23] differ for the implementation of the auxiliary PDs and the strategies adopted to scale the loop gain and the loop filter coefficients during the transient, however, overall, they fall within the same conceptual scheme. Unfortunately, a drawback of this approach is that in the quest for fast settling the frequency step driving the DCO must increase thus causing a larger residual frequency error left at the end of the transient, when the auxiliary loops switch off. If this error is too large to be recovered by the main-loop alone, the auxiliary paths are triggered again, leading to limit cycles highly degrading the PLL settling time (see Fig. 2). To remove this limitation, a novel type-II gear-shifting approach (GS-II) is proposed and implemented in this work.

The second approach is to use a frequency search algorithm to quickly set, in the event of a large frequency jump, the DCO frequency close to its final steady state value (see Fig. 2). To this aim, counters may be adopted to derive the DCO frequency corresponding to the minimum and maximum code driving the DCO and the code corresponding to the target frequency can be obtained from a linear interpolation [24]. Due to DCO nonlinearity a discrepancy will certainly exist

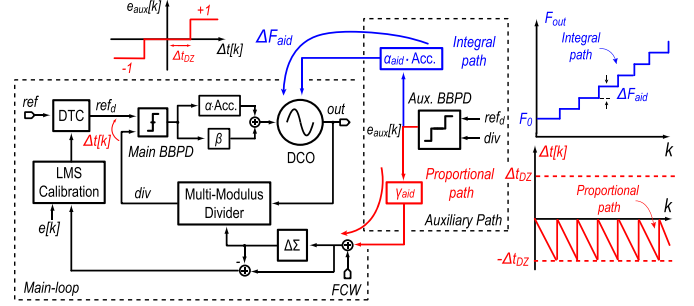


Fig. 3. Reference fractional- $N$  BBPLL architecture adopted in this work. Schematic transient of time error,  $\Delta t[k]$ , and DCO frequency,  $F_{out}$ , in response to a PLL frequency jump.

between the target and the actual frequency generated by the first correction, therefore the searching procedure can continue, with successive iterations of the linear interpolation until a specified error is achieved. In alternative, a binary search procedure [25] or a more computationally intensive approach [18] were also proposed. The limit of these solutions is the need for additional hardware and, when very accurate settling accuracy is needed, the number of iterations becomes too large. In this work, a new adaptive frequency switching (AFS) technique is introduced. Exploiting the fast settling performance due to the GS-II technique, the precision required to the initial DCO frequency setting is highly relaxed, thus avoiding iterations. In addition, the AFS implementation does not add hardware to the already existing BBPLL blocks. Both the GS-II technique and the AFS are implemented into a 28-nm CMOS BBPLL [9] with 8.5–10 GHz tuning range demonstrating a locking time better than  $1.56 \mu s$  (within 80 ppm accuracy) for frequency jumps up to 1.5 GHz while retaining a very low jitter at steady state.

This article is organized as follows. Section II describes the BBPLL architecture and the stability versus locking time trade-off met when pushing locking time using auxiliary frequency aid loops. The GS-II and AFS techniques are discussed in Section III together with their implementation into the BBPLL prototype. Section IV reports the measurements on the prototype, comparing the results with the state-of-the-art. Conclusions are drawn in Section V.

## II. BBPLL ARCHITECTURE

Fig. 3 describes the architecture of the fractional- $N$  BBPLL taken as a reference in this work, where an auxiliary path with a dead-zone  $\pm \Delta t_{dz}$  is introduced to achieve fast-locking. Anytime the PLL time error  $\Delta t[k]$  exceeds the time threshold  $\Delta t_{dz}$ , the auxiliary BBPD provides the sign function of this error. The proportional path of the auxiliary loop is implemented with a feedforward technique [7], adding the auxiliary BBPD output to the PLL frequency control word (FCW) after scaling it by a gain  $\gamma_{aid}$ . Fig. 3 (right) shows the transient of the time error,  $\Delta t[k]$ , triggered by a PLL frequency jump. Due to the proportional path, as the auxiliary BBPD threshold is crossed, the FCW is changed by  $\gamma_{aid}$ , causing a variation of the divided signal div period by

$$\Delta T_{aid} = \frac{\gamma_{aid}}{F_{out}} \quad (1)$$

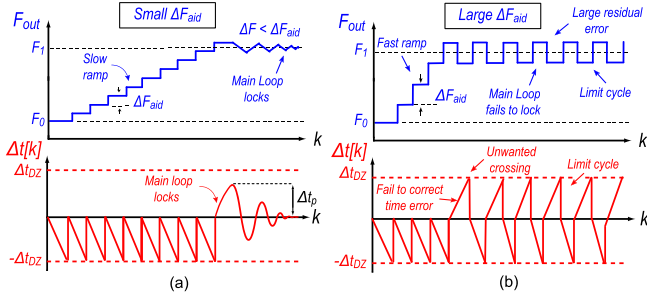


Fig. 4. Transient waveforms of the BBPLL loop with auxiliary path for different  $\Delta F_{\text{aid}}$  values: transient for (a) small values of  $\Delta F_{\text{aid}}$  and (b) large values of  $\Delta F_{\text{aid}}$  triggering the limit cycle.

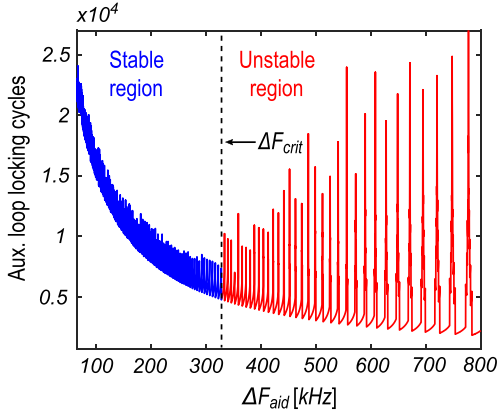


Fig. 5. Simulation results for the locking time of the auxiliary loop versus the parameter  $\Delta F_{\text{aid}}$  for a frequency jump of 20 MHz.

where  $F_{\text{out}}$  is the DCO frequency. In this way, by choosing  $\gamma_{\text{aid}} = F_{\text{out}} \Delta t_{\text{dz}}$ , the accumulated PLL time error is canceled as soon as the threshold  $\Delta t_{\text{dz}}$  is crossed (see Fig. 3). The time shift is highly accurate since the solution benefits from the presence of the high-resolution digital-to-time converter (DTC) already existing in the reference branch, which removes the quantization noise of the multimodulus divider (MMD) in fractional- $N$  mode [14]. The integral path of the auxiliary loop is instead implemented by integrating the auxiliary BBPD output with a gain  $\alpha_{\text{aid}}$  and feeding it to an additional DCO capacitor bank with a frequency resolution  $k_{f,\text{aid}}$ . As a consequence, any time the auxiliary BBPD threshold is crossed, the DCO frequency changes by

$$\Delta F_{\text{aid}} = \alpha_{\text{aid}} k_{f,\text{aid}} \quad (2)$$

as shown in Fig. 3 (right).

Fig. 4(a) depicts the overall frequency transient caused by a frequency jump large enough to trigger the auxiliary loop. The transient follows a staircase with a step equal to  $\Delta F_{\text{aid}}$  which lasts until the frequency error changes its sign and falls below  $\Delta F_{\text{aid}}$ . At this point the auxiliary BBPD threshold is not crossed anymore and the final PLL transient is driven by the main-loop only. This loop recovers the residual frequency error with a transient reaching a peak time error  $\Delta t_p$  [see Fig. 4(a)]. To speed up the initial frequency staircase, the  $\Delta F_{\text{aid}}$  value must be increased. Unfortunately also the residual frequency error left to the main-loop increases together with the corresponding peak time error  $\Delta t_p$ . If  $\Delta t_p$  exceeds  $\Delta t_{\text{dz}}$ ,

the auxiliary BBPD threshold is crossed again, but now in the opposite direction [see Fig. 4(b)] and the system falls into a limit cycle condition, where the auxiliary BBPD threshold is continuously crossed until, after a chaotic transient, the main-loop is eventually able to recover locking. This problem is exacerbated by the narrow bandwidth of the main-loop needed for PLL jitter minimization, which makes difficult to guarantee the condition  $\Delta t_p < \Delta t_{\text{dz}}$  for large residual frequency errors. In practice, to avoid limit cycles and the corresponding settling time degradation, the value of  $\Delta F_{\text{aid}}$  should be kept below a critical value  $\Delta F_{\text{crit}}$  that is expected to be limited by the narrow bandwidth of the main-loop. As a consequence, a trade-off between locking time and system stability arises, which is the price to pay to overcome the jitter versus locking time trade-off by adopting an auxiliary wide-bandwidth path with a coarse resolution PD.

The above intuitive discussion can be made quantitative by linking the value of  $\Delta F_{\text{crit}}$  to the parameters of the main loop and, in particular, to the PLL jitter performance. Notice that  $\Delta F_{\text{crit}}$  can be found as the minimum frequency error which causes the auxiliary loop to be triggered, i.e., the frequency error causing the peak time error  $\Delta t_p$  in Fig. 4(a) to exceed  $\Delta t_{\text{dz}}$ . This parameter was derived in [11] and [26] as

$$\Delta F_{\text{crit}} = \beta k_f + \sqrt{2R\beta k_f F_{\text{out}} F_{\text{ref}} \Delta t_{\text{dz}}} \quad (3)$$

where  $R = (\alpha/\beta)$  is the ratio between the integral and proportional parameters of the main loop shown in Fig. 3,  $k_f$  is the main-loop bit-to-frequency resolution of the DCO and  $F_{\text{ref}}$  is the reference frequency. As expected, (3) reveals that  $\Delta F_{\text{crit}}$  depends on  $\beta k_f$ , which, in turn, is proportional to the PLL bandwidth, usually set to minimize the PLL integrated jitter. The  $\beta k_f$  value leading to optimal PLL jitter is known in literature as [27], [28]

$$\beta_{\text{opt}} k_f = \sigma_{t,\text{dco}} F_{\text{out}}^2 \sqrt{\frac{F_{\text{ref}}}{F_{\text{out}}}} \quad (4)$$

where  $\sigma_{t,\text{dco}}$  is the DCO cycle-to-cycle jitter. Therefore, the DCO noise, which sets the PLL jitter performance, also sets the maximum value of  $\Delta F_{\text{crit}}$  via (3) and (4) and the achievable PLL locking time.<sup>1</sup> For example, in our design,  $F_{\text{out}} \approx 8.5$  GHz,  $F_{\text{ref}} = 250$  MHz,  $k_f \approx 300$  kHz and  $\sigma_{t,\text{dco}} \approx 1.7$  fs, obtained assuming a DCO phase noise of  $\mathcal{L}_{\text{dco}} = -120$  dBc/Hz at 1 MHz frequency offset and using [29]. With these values  $\beta_{\text{opt}} \approx 2^{-4}$ . Using  $R = 2^{-8}$  and  $\Delta t_{\text{dz}} = 200$  ps, from (3)  $\Delta F_{\text{crit}}$  turns out to be only 270 kHz.<sup>2</sup> To derive the corresponding limitation on locking time, a behavioral simulation of the loop with the same parameters was performed. Fig. 5 shows the results for a

<sup>1</sup>The trade-off between jitter and locking time is more critical in PLLs with LC-based DCOs where the DCO phase noise can be several tens of dB smaller than that of ring-based DCOs.

<sup>2</sup>The  $R$  value affects the frequency of the zero in the loop gain. By changing  $R$ , the zero can be properly set not only to get a safe phase margin but also to optimize the jitter, filtering off the close-in flicker phase noise of the DCO. In our design the value  $R = 2^{-8}$  was due to noise optimization. The corresponding phase margin, computed following the linear analysis proposed in [30] turns out to be of about  $75^\circ$ , taking into account that the main-loop is characterized by  $D = 4$  latencies and an input-referred jitter of approximately  $\sigma_r = 210$  fs.

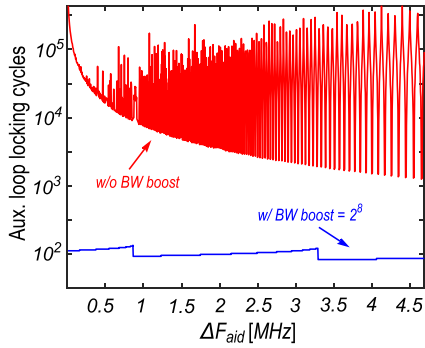


Fig. 6. Locking time of the auxiliary loop simulated as a function of the parameter  $\Delta F_{\text{aid}}$  with and without the adoption of a bandwidth boost by a factor  $2^8$ . The simulations refer to a frequency jump of 200 MHz.

frequency jump of 20 MHz. The locking time of the auxiliary path, defined as the number of reference cycles needed for the auxiliary loop to become idle, improves as  $\Delta F_{\text{aid}}$  increases, as expected. However, for  $\Delta F_{\text{aid}} > 320$  kHz, the locking time starts to show some peaks due to limit cycles, which become more frequent and with an increasing amplitude as  $\Delta F_{\text{aid}}$  increases. It is noted that (3) can be used to approximate the crossover between the stable and unstable locking regions. Due to limit cycles, the locking time cannot be pushed below 5000 reference cycles, thus posing a strong limitation to the settling performance of this scheme, which become even worse at larger frequency jumps.

### III. PROPOSED FAST-LOCKING TECHNIQUES

#### A. Limit Cycle Avoidance Through Gear-Shifting

Equation (3) suggests to increase  $\Delta F_{\text{crit}}$  by boosting the parameter  $\beta$ , while keeping the same value of  $R$ . This solution corresponds to broaden the loop bandwidth speeding-up the locking transient. For instance, with the parameters of our design, reported in Section II, a boosting factor of  $2^8$  on both  $\alpha$  and  $\beta$  would shift  $\Delta F_{\text{crit}}$  to 8.8 MHz, preventing limit cycles and enabling the use of a larger  $\Delta F_{\text{aid}}$  to shorten the PLL locking time. Fig. 6 compares the simulated locking time of the auxiliary path by increasing  $\Delta F_{\text{aid}}$  with and without the adoption of the bandwidth boost, for a frequency jump of 200 MHz. Even limiting the maximum value of  $\Delta F_{\text{aid}}$  to  $\Delta F_{\text{crit}}/2$  to guarantee some margin, the locking time can be reduced by more than two orders of magnitude with this approach. On the other hand, to circumvent the degradation of the PLL jitter at steady state, the bandwidth boost can only be temporary and must fade out as locking is approaching. To this aim, a proper *gear-shift* procedure [31] should be implemented to bring the loop filter coefficients from their boosted values ( $\alpha_{\text{gs}}, \beta_{\text{gs}}$ ) back to those optimal for jitter minimization, ( $\alpha_{\text{opt}}, \beta_{\text{opt}}$ ). Fig. 7(a) shows a practical implementation of the gear-shift scheme. The auxiliary BBPD output is monitored. Anytime its threshold is crossed, signifying that the PLL is out-of-lock, and that a limit cycle may be triggered, the loop filter coefficients are boosted to  $\alpha_{\text{gs}}$  and  $\beta_{\text{gs}}$ . A lock-detector circuit, based on [13], computes the running average of the main BBPD signal  $e[k]$  over  $m$  reference cycles, 32 in our system. Anytime the average value falls below a threshold

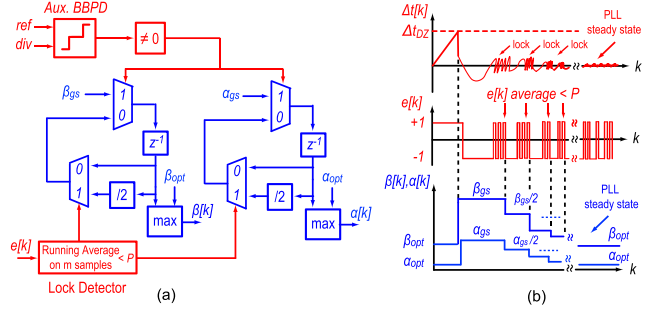


Fig. 7. Gear-shift procedure to avoid limit cycles: (a) block diagram of the implemented gear-shift module and (b) gear-shift waveforms.

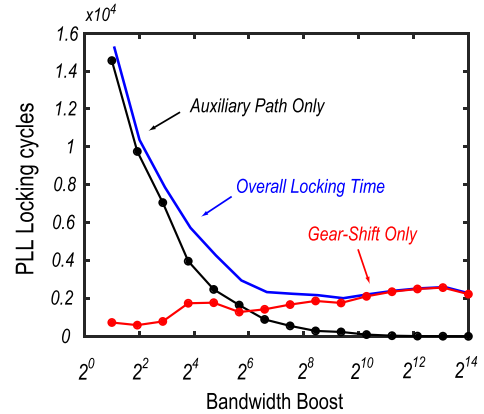


Fig. 8. Dependence of the PLL locking time on the bandwidth boost factor applied to both loop filter parameters ( $\alpha, \beta$ ) while keeping constant their ratio,  $R$  to the steady state value  $2^{-8}$ . The simulations refer to a frequency jump of 400 MHz. The figure also shows the two components of the settling time: the time spent with the auxiliary path active and the time needed to settle the main-loop using the gear-shift procedure described in the text.

$P$ , set to  $1/8$  in this design, both coefficients are divided by a factor of 2 until, step by step, they seamlessly reach their steady state values, as schematically illustrated in Fig. 7(b). Such a gradual procedure is preferable with respect to an abrupt switch to their final values, as a limit cycle may be triggered again due to the residual frequency error still existing at the bandwidth switching instant. Section III-C deals with the design guidelines followed to choose the appropriate values of  $P$ ,  $m$ , and the scaling factor of the loop filter coefficients involved during the gear-shift.

Note that, although a higher  $\Delta F_{\text{aid}}$  value improves the locking time of the auxiliary path, this comes at the cost of a larger number of gear-shifting steps, since a wider bandwidth boost would be necessary to suppress limit cycles. In practice, a too large bandwidth boost would result into a time-consuming gear-shifting sequence, thus impairing the PLL locking time. This trade-off is highlighted in Fig. 8, which shows the simulated PLL locking time, defined as the number of reference cycles needed to recover the optimal values ( $\alpha_{\text{opt}}, \beta_{\text{opt}}$ ), as a function of the bandwidth boost for a frequency jump of 400 MHz. In the simulation, for each value of the bandwidth boost,  $\Delta F_{\text{aid}}$  was set to the corresponding  $\Delta F_{\text{crit}}/2$  obtained using (3). As the bandwidth boost increases, the locking time of the auxiliary path improves while the time needed for the gear-shift sequence to settle becomes larger.

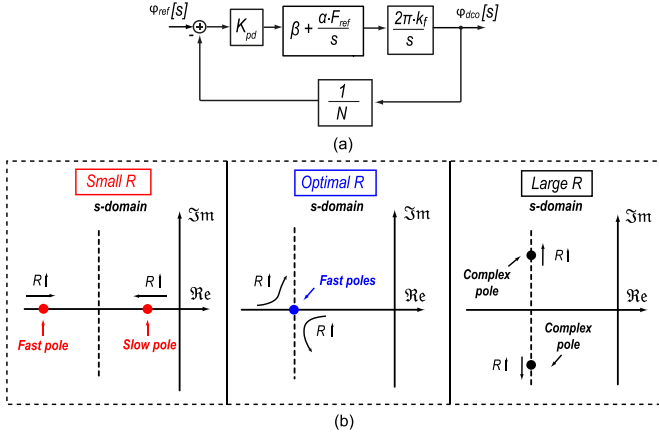


Fig. 9. Linear PLL analogy: (a) linear  $s$ -domain phase model of a PLL, where  $\varphi_{ref}(s)$  and  $\varphi_{dco}(s)$  are the phase of the reference and DCO signals, respectively, and (b) dependence of the closed-loop poles on the ratio  $R = \alpha/\beta$ .

### B. Type-II Gear-Shift

Note that the gear-shift procedure described above boosts  $\alpha$  and  $\beta$  by the same factor, thus retaining the same  $R = 2^{-8}$  ratio needed to optimize steady state performance. However, one might wonder if a different  $R$  ratio should be adopted during the transient to minimize the settling time. To gain some preliminary insight, it is worth referring to the dependence on  $R$  of the PLL closed-loop singularities as derived from a PLL linear model. Despite the strong nonlinearity of the actual BBPLL system, the analysis is still useful as an analogy to interpret the results of behavioral simulations. The input to output transfer function of the scheme in Fig. 9(a) is given by

$$H(s) = \frac{\varphi_{dco}(s)}{\varphi_{ref}(s)} = N \frac{K R F_{ref} + K s}{s^2 + K s + K R F_{ref}} \quad (5)$$

where  $K = 2\pi K_{pd} k_f \beta / N$ . Fig. 9(b) shows the position of the closed loop poles in the complex plane as a function of  $R$ . For  $R < K T_{ref}/4$ , where  $T_{ref}$  is the reference period, the closed-loop poles are real. A low-frequency pole is close to the imaginary axis thus causing a slow PLL response. By increasing  $R$ , this pole moves at higher frequencies thus speeding-up the PLL transient. For  $R = K T_{ref}/4$ , the poles become real and coincident, reaching the maximum distance from the imaginary axis and leading to the minimal locking time. A larger value of  $R$  leads to a complex-conjugate pair. In this regime, by increasing  $R$  the real part remains the same but the imaginary part increases. The PLL phase margin degrades while oscillations start to appear in the response, meaning that  $R = K T_{ref}/4$  is the optimal choice for settling time. The trends suggested by the linear analysis and the presence of an optimal  $R$  value for settling hold also when considering the nonlinear BBPLL system together with the gear-shift technique introduced in Section III-A. To verify this property, behavioral simulations of the system were performed considering the operation of the main loop in Fig. 3 and the gear-shift scheme in Fig. 7(a) only. In this simulation, the auxiliary loop was supposed to be triggered the last time before becoming idle and therefore the gear-shift procedure was set at its initial step, i.e.,  $\alpha = \alpha_{gs}$  and  $\beta = \beta_{gs}$ , while an initial frequency error of 5 MHz (i.e., around  $\Delta F_{crit}/2$  for a

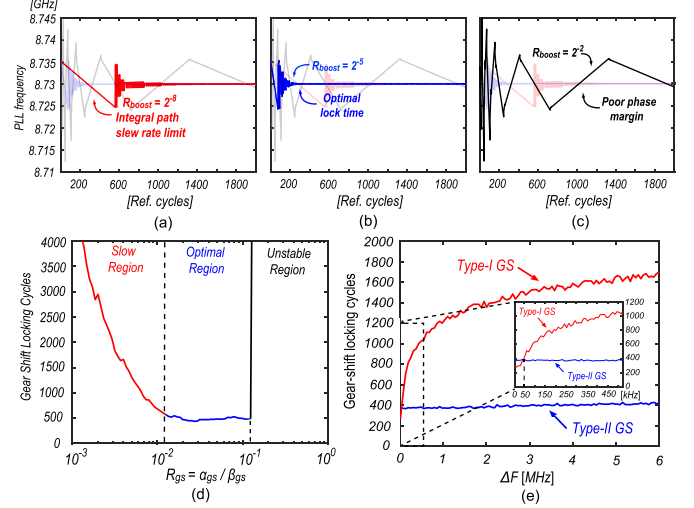


Fig. 10. Type-II gear-shift: behavioral simulation results of the settling time for an initial frequency error  $\Delta F = 5$  MHz using  $\beta_{gs} = 2^4$  and  $\alpha_{gs} = R_{gs} \cdot \beta_{gs}$ , with (a)  $R_{gs} = 2^{-8}$ , (b)  $R_{gs} = 2^{-5}$ , (c)  $R_{gs} = 2^{-2}$ , (d) variable  $R_{gs}$ , and (e) settling time comparison of type-I ( $R_{gs} = 2^{-8}$ ) and type-II ( $R_{gs} = 2^{-5}$ ) gear-shift as a function of  $\Delta F$ .

bandwidth boost of  $2^8$ ) was enforced. The value of  $\beta_{gs}$  was set to  $2^4$ , i.e., a boosting of  $2^8$  from  $\beta_{opt} = 2^{-4}$ , while  $\alpha_{gs}$  was set to  $R_{gs} \cdot \beta_{gs}$ , with variable  $R_{gs}$ , to investigate the locking time dependence on  $R_{gs}$ .<sup>3</sup> Fig. 10(d) shows the results. For  $R_{gs}$  values smaller than about  $10^{-2}$  the integral gain of the loop filter is too small, thus causing a slow transient due to the limited frequency update rate [see Fig. 10(a)]. On the other hand,  $R_{gs}$  values larger than about  $10^{-1}$  correspond to a poor phase margin of the closed-loop system, resulting into large oscillations [see Fig. 10(c)]. The system is on the verge of instability and the locking time increases. For  $R_{gs}$  values ranging from  $10^{-2}$  to  $10^{-1}$  optimal locking performance can be achieved [see Fig. 10(b)] almost independently of  $R_{gs}$ . Note that this property is inherited from the linear PLL in Fig. 9(a), where the real part of the closed-loop poles and so the PLL locking time remain constant for  $R > K T_{ref}/4$ . The procedure described below differs from a conventional gear-shift, as it does not only boost  $\beta$  for bandwidth broadening but also  $R$  for speeding-up the transient. For this reason, we denoted it as GS-II, while the conventional gear-shift can be denoted as *type-I gear shift* (GS-I). The gear-shift scheme in Fig. 7(a) was designed to boost the values of  $\alpha$  and  $\beta$  to  $\beta_{gs} = 2^4$  and  $\alpha_{gs} = 2^{-1}$ , corresponding to  $R = 2^{-5}$ , right after the auxiliary BBPD threshold crossing. During the locking transient, based on the running average of the main BBPD signal, the two coefficients are divided by a factor of 2 at each step still keeping  $R = 2^{-5}$  until  $\beta$  reaches the steady state value of  $2^{-4}$ . At this point  $\alpha_{gs}$  is still  $2^{-9}$ . In the last steps of the gear-shift procedure  $\alpha_{gs}$  is therefore lowered down to  $2^{-12}$ , thus recovering the loop filter coefficients for minimum jitter. Since  $R = 2^{-5}$  lies within the optimal region of Fig. 10(d), the main-loop locking time is minimized. Note that the locking time advantage of the GS-II technique is

<sup>3</sup>In this simulation, the locking time was computed as the time needed to achieve the settling of the loop filter coefficients through the gear-shift procedure, i.e.,  $\alpha = \alpha_{opt}$  and  $\beta = \beta_{opt}$ .

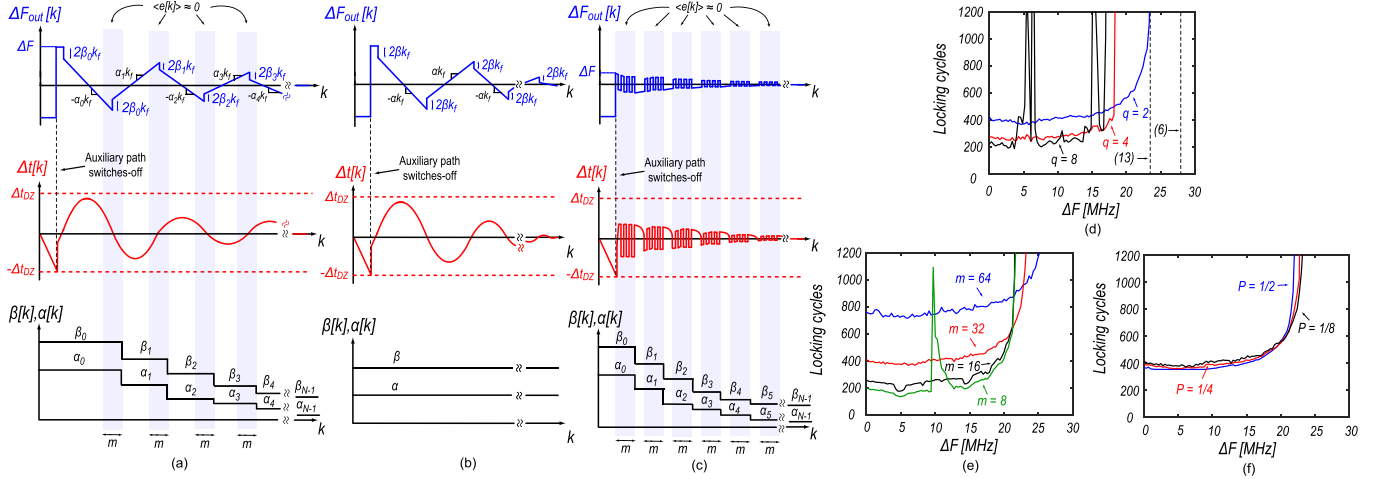


Fig. 11. Transient waveforms of the PLL frequency error  $\Delta F_{out}[k]$ , the time error  $\Delta t[k]$ , and loop-filter gains  $\beta[k]$  and  $\alpha[k]$  for (a) BBPLL with gear-shift, (b) conventional BBPLL, (c) BBPLL with gear-shift with  $\Delta F \ll \Delta F_{max}$ , and simulated locking time at different values of (d) gear-shift scaling factor  $q$ , (e) running average length  $m$ , and (f) average threshold  $P$ .

consistently achieved over a wide range of initial frequency errors  $\Delta F$ , i.e., the frequency error existing at the start of the gear-shift procedure. Fig. 10(e) indeed compares the simulated GS-I and GS-II locking times as a function of  $\Delta F$ , showing that GS-II outperforms GS-I almost independently of  $\Delta F$ , with their performance becoming comparable only for unpractically small values, i.e., below 50 kHz in our design.

Equation (3) can be used to illustrate another benefit of the proposed technique. By boosting the  $R$  parameter, the corresponding  $\Delta F_{crit}$  value increases, allowing to either adopt a larger  $\Delta F_{aid}$  to further speed-up the auxiliary path locking transient or, by keeping the same  $\Delta F_{aid}$ , it provides additional margins to avoid limit cycles. In our design, the boosted  $R$  value of  $2^{-5}$  shifts  $\Delta F_{crit}$  from 8.8 MHz to about 16 MHz. We have therefore adopted  $\Delta F_{aid} \approx 6$  MHz exploiting the additional design margins.

### C. Gear-Shift Parameters Impact on Locking Transient

To understand how to properly design the values of the loop-filter coefficients scaling factor, the length of the running average  $m$  and the average threshold value  $P$  used during the gear-shift, their effect on the BBPLL dynamic should be investigated. Fig. 11(a) illustrates the transient waveforms of the PLL frequency error  $\Delta F_{out}[k]$ , the time error  $\Delta t[k]$  and the loop filter gains  $\beta[k]$  and  $\alpha[k]$  during the gear-shift, where an initial frequency error  $\Delta F$  is supposed to be left by the auxiliary path after becoming idle. The residual transient resembles the one of a conventional BBPLL (i.e., without gear-shift) shown in Fig. 11(b), where the frequency error follows a triangularly shaped pattern with alternating slopes of  $\pm \alpha k_f$  and being reduced by  $2\beta k_f$  anytime a sign change of  $e[k]$  occurs [12]. However, in the BBPLL with gear-shift, the values of  $\alpha$  and  $\beta$  in each transient section of Fig. 11(a) are different. As the  $e[k]$  running average  $\langle e[k] \rangle$  drops below the threshold  $P$  near the  $\Delta t[k]$  sign inversions, the loop filter coefficients scale down, as depicted in Fig. 11(a). Therefore, the values of  $\alpha$  and  $\beta$  at the  $n$ th transient section can be, respectively, derived as  $\alpha_{gs}/q^n$  and  $\beta_{gs}/q^n$ , where  $q$  is the chosen gear-shift scaling

factor. To achieve locking within the  $N$  available gear-shift steps,  $\Delta F$  should be smaller than a maximum value  $\Delta F_{max}$  that can be derived as the sum of the individual frequency error correction terms provided by the BBPLL at each transient section (i.e.,  $2\beta_n k_f$ ). It is<sup>4</sup>

$$\Delta F_{max} = 2\beta_0 k_f + \sum_{n=0}^{N-1} 2\beta_n k_f \approx 2\beta_{gs} k_f \frac{2q-1}{q-1}. \quad (6)$$

To pursue a robust operation while simplifying the gear-shift implementation, the value of  $q$  was set to maximize  $\Delta F_{max}$ , based on (6), under the constraint of being a power of two,<sup>5</sup> resulting in the choice of  $q = 2$  in this design.

Let us now discuss how to choose the running average length  $m$ . To this aim, it is useful to analyze the gear-shift behavior for  $\Delta F$  well below  $\Delta F_{max}$  [see Fig. 11(c)]. If the initial frequency error is small, the PLL is able to recover during the first gear-shift step. However, the values of  $\alpha$  and  $\beta$  need to reach the steady state values, requiring at least

$$M_{gs,min} = m \cdot N. \quad (7)$$

additional clock cycles, as the  $\langle e[k] \rangle$  computation requires at least  $m$  new  $e[k]$  samples at each gear-shift step.<sup>6</sup> From (7), a smaller  $m$  should result in a locking time reduction. However, small  $m$  values would hinder the gear-shift robustness. The limiting case is  $m = 2$ , when the gear-shift scaling is triggered whenever  $e[k]$  changes its sign, thus making the procedure sensitive to noise and disturbances. To avoid this problem, the value of  $m = 32$  was chosen in this design to produce an average over a significant part of the BBPLL transient.<sup>7</sup> The value of  $P$  was likewise chosen

<sup>4</sup>The additional term  $2\beta_0 k_f$  in (6) is due to  $\beta[k]$  already being equal to  $\beta_0 = \beta_{gs}$  before the auxiliary path switch-off, as a result of the multiple threshold crossings occurred in the preceding transient.

<sup>5</sup>In this case the scaling operation can be implemented as a simple bit-shift.

<sup>6</sup>When compared to GS-I, GS-II more easily achieves the minimum  $M_{gs,min}$  locking cycles, as its larger boost of  $\alpha[k]$  allows to quickly push  $\langle e[k] \rangle$  again to zero after the loop-filter coefficients scaling event.

<sup>7</sup>Taking  $\Delta F = \Delta F_{crit} \approx 16$  MHz as a worst case, each transient section of Fig. 11(a) is smaller than about 80 cycles and  $m$  was chosen as a power of two close to half this value, i.e.,  $m = 32$ .

to favor gear-shift robustness. Note that, since the minimum variation of  $\langle e[k] \rangle$  is  $2/m$ ,<sup>8</sup> if  $P < 2/m$  the gear-shift scaling would be triggered only when  $\langle e[k] \rangle = 0$ , which could take a long time before being exactly obtained. On the other hand, a too large value of  $P$  would hinder the average effectiveness. The limiting case is  $P = 1$ , meaning that the gear-shift scaling would be triggered independently of the measured  $\langle e[k] \rangle$  (since  $\langle e[k] \rangle \leq 1$  always holds). As a compromise, in this design,  $P$  was chosen to be  $4/m = 1/8$ , i.e., a factor of 2 larger than the minimum significant value.

Fig. 11(d)–(f) shows the simulated GS-II locking time dependence on the initial frequency error  $\Delta F$  by varying  $q$ ,  $m$  and  $P$ , respectively. In Fig. 11(d), the value of  $\Delta F$  causing the locking time to substantially increase, i.e.,  $\Delta F_{\max}$ , is maximized for  $q = 2$ , as expected.<sup>9</sup> The prediction of  $\Delta F_{\max}$  based on (6), however, overestimates the simulated value. This discrepancy is caused by the frequency perturbations induced by the loop-filter coefficients scaling and a more precise estimate is derived in Appendix A (13). Fig. 11(d) also shows that for  $q = 4$  a better locking time can be achieved, as the required number of gear-shift steps  $N$  reduces<sup>10</sup> at the expense of a smaller  $\Delta F_{\max}$ , while for larger values of  $q$  the bandwidth scaling applied by the gear-shift is so abrupt that the BBPLL is not able to recover, causing the large overshoots in Fig. 11(d) for  $q = 8$ . Fig. 11(e) verifies that the locking time can be improved by reducing  $m$ , however, for  $m = 8$  the running average length is so short that, even without any external disturbance, the locking time exhibits a large overshoot, caused by the poor averaging operation.<sup>11</sup> The simulated locking time dependence on  $P$ , shown in Fig. 11(f), is the weakest among the above parameters, meaning that smaller values of  $P$  should be preferred (provided that  $P > 2/m$ ) as they improve gear-shift robustness without impairing settling performance.

#### D. Adaptive Frequency Switching

The GS-II unit is intended to minimize the settling time of the main-loop. However, for large frequency jumps, the overall settling time may be still limited by the initial part of the transient when the auxiliary loop is active and the DCO frequency is changed by  $\Delta F_{\text{aid}}$  per step, following a ramp to reach the steady state value. This phase could be skipped, or at least significantly shortened, if the PLL frequency can be quickly switched to a coarse estimate of its final value, by acting on an additional coarse DCO capacitor bank. When the PLL is locked to a frequency  $F_0$  and a frequency switching has to be performed, the amplitude of the frequency jump is known a priori, being equal to  $\Delta \text{FCW} \cdot F_{\text{ref}}$ , where  $\Delta \text{FCW}$  is the wanted variation of the PLL FCW. However, since the DCO tuning curve is nonlinear and affected by process, voltage

<sup>8</sup>This follows by computing the  $\langle e[k] \rangle$  variation caused by a sign change of a single  $e[k]$  sample within the average window.

<sup>9</sup>As  $\Delta F_{\max} \approx 24$  MHz,  $\Delta F_{\text{crit}} \approx 16$  MHz sets the most stringent constraint on  $\Delta F$ , however, maximizing  $\Delta F_{\max}$  is still useful to have some margin.

<sup>10</sup> $N = \log_q(\alpha_{\text{gs}}/\alpha_{\text{opt}})$ , where  $\alpha$  is used in this formula, rather than  $\beta$ , as it experiences the largest boost due to GS-II.

<sup>11</sup>Although Fig. 6 shows that an  $m$  value equal to 16 provides a better locking time, the value 32 was still chosen to improve robustness, thanks to the larger margin taken from the simulated critical value of 8 and the longer gear-shift averaging time.

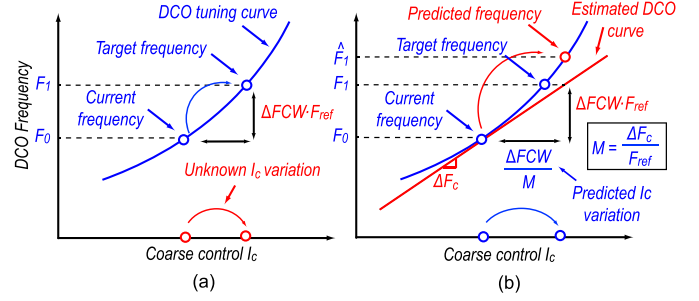


Fig. 12. AFS technique: (a) schematic representation of the DCO tuning curve and problem of DCO coarse control code variation estimation, and (b) linear approximation of the DCO tuning curve through its tangent line and corresponding prediction of the coarse control code variation using the parameter  $M = \Delta F_c / F_{\text{ref}}$ .

and temperature (PVT) spreads, the corresponding variation of the coarse DCO bank control code  $I_c$  bringing the output frequency close to its final target is instead unknown [see Fig. 12(a)]. To solve this problem, the idea was to derive at runtime an estimate of the local slope of the DCO tuning curve, denoted as  $\Delta F_c$  in Fig. 12(b), which corresponds to the bit to frequency gain of the coarse DCO bank. In this way, a linear estimate of the DCO tuning curve can be derived using the tangent line intersecting the tuning curve at the current frequency  $F_0$ , as depicted in Fig. 12(b), and the variation of the coarse control code can be predicted as

$$\Delta I_c = \frac{\Delta \text{FCW} \cdot F_{\text{ref}}}{\Delta F_c} = \frac{\Delta \text{FCW}}{M} \quad (8)$$

where  $M = \Delta F_c / F_{\text{ref}}$  is unknown. To produce a runtime estimate of  $M$ , before driving the PLL with a frequency jump, the system enters into an *estimation mode*. The DCO coarse control code  $I_c[k]$  is increased by one for one reference cycle, thus making the PLL output frequency increasing by  $\Delta F_c$  in the same clock cycle [see Fig. 13(b)]. Such a frequency perturbation injects a time error  $\Delta t_c$  between the  $\text{ref}_d$  and  $\text{div}$  signals at the BBPD input, which depends on  $M$  as

$$\Delta t_c \approx \frac{\Delta F_c}{F_{\text{ref}}} \cdot T_0 = M \cdot T_0 \quad (9)$$

where  $T_0$  is the DCO period before the frequency perturbation is applied. In principle  $\Delta t_c$  could be measured with a high-resolution and linear TDC and the value of  $M$  can be derived from the TDC output code upon the knowledge of the TDC gain. However, accurate and linear TDCs are difficult to implement. In alternative, the DTC already existing inside the main-loop can be exploited. To this aim, a digital calibration signal  $\text{cal}[k]$ , which is a digital staircase [see Fig. 13(c)], is generated and added to the input of the DTC, before the multiplication by the least mean square (LMS) gain inside the LMS calibration block [see Fig. 13(a)]. Notice that, thanks to the LMS algorithm, the DTC gain is known and forced to be equal to the DCO period  $T_0$  [14], which is an advantage with respect to the use of an auxiliary TDC. Additionally, the DTC is generally already designed to be highly linear to suppress fractional spurs. Following this procedure, at each step a time error equal to  $T_0 \cdot \text{cal}[k]$  is removed from the main-loop,<sup>12</sup>

<sup>12</sup>This follows from the DTC gain having a magnitude equal to  $T_0$  and a negative sign in the scheme in Fig. 13(a).

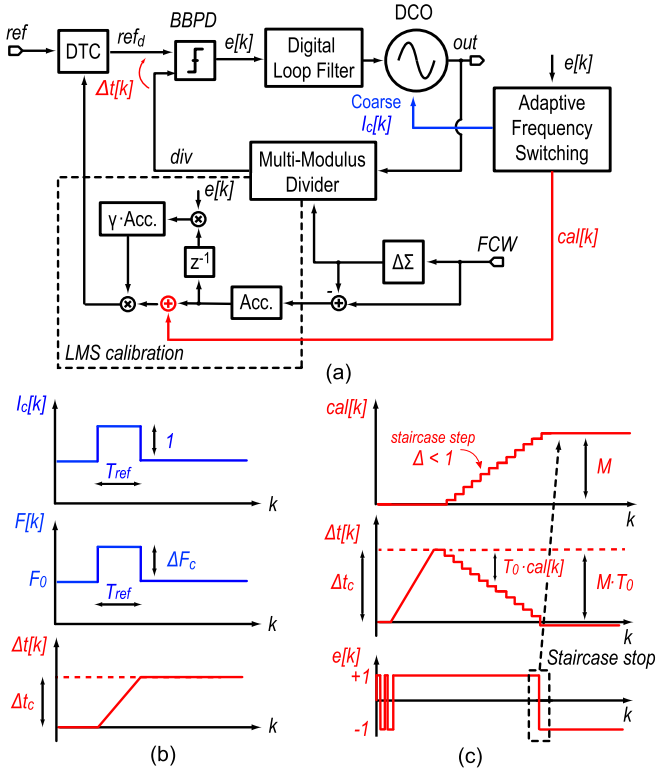


Fig. 13. AFS technique. (a) Block scheme implementing the measurement of the parameter  $M$ . (b) When in *estimation mode* the AFS block increments by one the DCO coarse control code,  $I_c[k]$ , for one reference cycle, injecting a time error  $\Delta t_c$  in the PLL loop. (c) Waveforms of  $cal[k]$ ,  $\Delta t[k]$ , and  $e[k]$  during the measurement of  $M$  by exploiting the DTC.

meaning that the initial injected time error  $\Delta t_c$  is progressively canceled as the  $cal[k]$  signal increases [see Fig. 13(c)]. The AFS block monitors the main BBPD signal  $e[k]$  and stops the staircase signal  $cal[k]$  as soon as a sign change is detected, as shown in Fig. 13(c). This crossover condition corresponds to complete time error removal and the final value reached by  $cal[k]$  provides the desired estimate of  $M$ . As a matter of fact, this procedure implements a frequency-to-digital converter by performing a linear search for the value of  $M$  to produce an estimate of the coarse DCO gain  $\Delta F_c$ . The frequency resolution  $\Delta F_{res}$  of this estimate depends on the staircase step, denoted as  $\Delta$  in Fig. 13(c). As  $\Delta$  sets the precision on measuring  $\Delta t_c$ , the corresponding frequency resolution  $\Delta F_{res}$  is obtained from (9) as

$$\Delta F_{res} = \frac{\Delta \cdot F_{ref}}{T_0}. \quad (10)$$

The number of clock cycles needed to perform the  $\Delta t_c$  measurement can be instead derived as

$$N_{meas} = \frac{\Delta t_c}{\Delta} = \frac{\Delta F_c}{\Delta F_{res}} \quad (11)$$

where, in the last step, (9) and (10) were used. The above equation highlights a trade-off between accuracy and measurement time, which is a typical trait of a linear search. Notice that the worst case measurement time occurs when, due to DCO nonlinearity and PVT variations,  $\Delta F_c$  assumes its maximum value  $\Delta F_{c,max}$ . Given  $\Delta F_{c,max}$  and the maximum number of

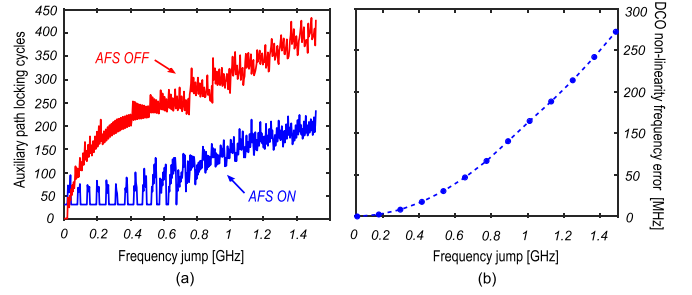


Fig. 14. AFS technique. (a) Comparison between the simulated locking time of the auxiliary path with and without the AFS as a function of the frequency jump. (b) Frequency errors induced by the DCO nonlinearity.

cycles allocated for the AFS estimation, the minimum resolution is derived. In this design, to not substantially degrade the PLL locking time,  $\Delta F_{res}$  was chosen in such a way to allocate a maximum of 50 cycles for the AFS estimation. Taking into account that  $\Delta F_{c,max} < 80$  MHz, as derived from circuit-level simulations, the corresponding frequency resolution turns out to be 1.5 MHz.

Note that any block acting on the time error  $\Delta t[k]$  during the AFS estimation phase can, in principle, affect its accuracy. For instance, after the AFS injection of  $\Delta t_c$ , the main loop senses the time error and reacts to reduce it. Similar perturbations of  $\Delta t[k]$  could in principle be induced by the DTC gain calibration loop, or by the auxiliary path if accidentally turned on by the  $\Delta t_c$  injection. In our system, the above loops do not significantly impair the AFS accuracy<sup>13</sup>; however, for designs where these effects may be critical,<sup>14</sup> they can be simply switched off during the AFS estimation phase by forcing to zero the error signals  $e[k]$  and  $e_{aux}[k]$  fed to the loop-filters and LMS calibration block. In this way, their operation is temporarily frozen until the end of the estimation phase.

Once  $M$  is obtained, the predicted variation of the coarse control code is derived using (8). Despite linear estimates of the DCO tuning curve were already exploited in [18], [25], and [24], the advantage of the AFS technique is that this approximation is obtained without adding significant hardware to the already existing PLL architecture. Additionally, the residual frequency errors caused by DCO nonlinearity are quickly corrected by the auxiliary loop together with the GS-II unit, and therefore no further iteration of the linear estimate are implemented. Fig. 14(a) shows the simulated locking time of the auxiliary path for negative frequency jumps up to 1.5 GHz starting from 10 GHz, with and without AFS technique. At small frequency jumps, the AFS locking time is almost constant, being in principle only limited by the duration of the DCO gain estimation phase. The fluctuations are caused by truncation errors when rounding the estimated

<sup>13</sup>The maximum  $\Delta t_c$  value, based on (9), is below 40 ps, thus not triggering the 200 ps auxiliary path dead-zone. Moreover, being  $e[k]$  constantly equal to 1 during the AFS estimation, its correlation with the fractional- $N$  quantization noise is zero, hardly producing any variation of the DTC gain. The main-loop instead, due to its narrow bandwidth, was verified to take more than 4000 cycles to recover from the maximum injected  $\Delta t_c$ , meaning that its contribution is negligible during the short 50 cycles AFS estimation phase.

<sup>14</sup>For instance, in a wide bandwidth PLL the perturbation induced by the main-loop may degrade AFS accuracy.



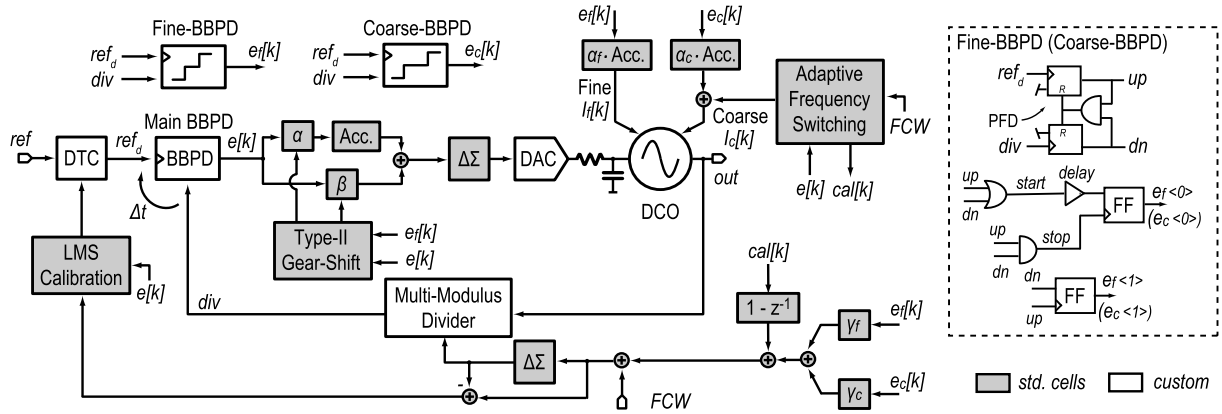


Fig. 15. Block diagram of the implemented system. Two auxiliary BBPDs are adopted to quickly handle large frequency errors. The AFS calibration signal  $cal[k]$  is differentiated and then added to the frequency control world, preventing the extension of the DTC range.

$\Delta I_c$  from (8) to the closest integer representing the number of capacitors to be switched within the coarse DCO bank.<sup>15</sup> At larger frequency jumps, instead, the AFS locking time starts to increase, mainly due to the residual frequency errors caused by DCO nonlinearity. With reference to the prototype DCO curve depicted in Fig. 12(b), nonlinearity causes the AFS to switch the PLL frequency from  $F_0$  to  $\hat{F}_1$  departing from the actual target frequency  $F_1$  that instead lies on the tangent line approximation of the DCO curve passing from  $F_0$ . Fig. 14(b) shows the frequency error induced by the nonlinearity of the implemented DCO, i.e.,  $\hat{F}_1 - F_1$ , as a function of the frequency jump, i.e.,  $F_1 - F_0$ , for negative jumps starting from  $F_0 = 10$  GHz. As expected, as the frequency jump increases, the frequency error becomes larger, explaining the longer AFS locking time at larger frequency jumps in Fig. 14(a). Even considering the above limitation, the AFS technique is an efficient way to reduce the auxiliary path locking time.<sup>16</sup>

When compared with more complex DCO tuning curve estimation schemes based on adaptive lookup tables (LUTs) [19], [32], [33], which can even track DCO nonlinearity, the AFS technique avoids the use of frequency modulation training sequences, thus enabling its adoption even when the PLL generates an unmodulated carrier. Furthermore, by avoiding to wait the convergence of the LMS algorithms needed to fill the LUT registers, which can take several hundreds of microseconds [33], the tuning curve estimation provided by the AFS unit, despite being less accurate, is inherently faster.<sup>17</sup>

#### IV. IMPLEMENTATION AND MEASUREMENTS

Fig. 15 shows a block diagram of the implemented PLL, characterized by a conventional fractional- $N$  main BBPLL loop which operation is aided by the GS-II module and the

<sup>15</sup>This is also the reason why the division operation in (8) has low hardware complexity, as all fractional bits of the result can be truncated.

<sup>16</sup>Only for small frequency jumps below about 20 MHz, which are not critical for determining the performance over the tuning range, the AFS shows a disadvantage, as the locking time is already so small that it falls below the AFS estimation phase duration.

<sup>17</sup>The waiting time needed for the convergence of LMS algorithms can be skipped by operating them only in foreground. However, in this case, environmental variations would not be tracked.

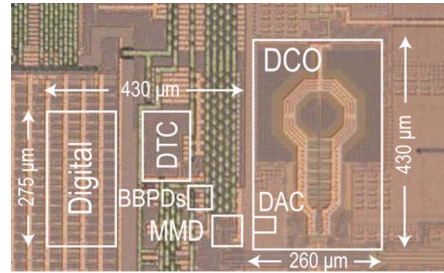


Fig. 16. Die micrograph.

AFS block. The auxiliary path is implemented using two nested auxiliary loops (based on the fine and coarse BBPDs), with progressively larger dead-zones equal to 200 and 400 ps, respectively. Once activated, the integral part of these loops drive the fine and the coarse DCO banks, with gains set to shift the DCO frequency by about 6 and 40 MHz, respectively.<sup>18</sup> Each of the auxiliary BBPD was implemented with a start-stop TDC scheme, also shown in Fig. 15, where the dead-zone is obtained by introducing a delay on the *start* signal, and the overall two bit error signal is derived by combining the two flip-flops (FF) outputs, with the most significant bit (MSB) and least significant bit (LSB) representing the error sign and magnitude (denoted with the suffix  $\langle 1 \rangle$  and  $\langle 0 \rangle$  in Fig. 15), respectively. Being implemented as a delay, the dead-zone value suffers from PVT spreads therefore affecting  $\Delta F_{crit}$ , as expected from (3). However, thanks to the design margins ensured by choosing  $\Delta F_{aid}$  much smaller than  $\Delta F_{crit}$  in the nominal conditions (as discussed in Section III-B), PVT spreads are not an issue. Appendix B provides a more detailed discussion on the operation of the auxiliary path upon PVT spreads of the dead-zone value.

It may be noticed that, in the final implementation of the system in Fig. 15, the AFS calibration signal  $cal[k]$  is first differentiated and then added to the PLL FCW, rather than being directly added at the DTC input. This scheme results

<sup>18</sup>Being implemented as capacitor banks, the gains of the fine and coarse DCO banks are nonlinear and affected by PVT variations. These values are computed close to the center of the DCO tuning curve and in the nominal conditions.

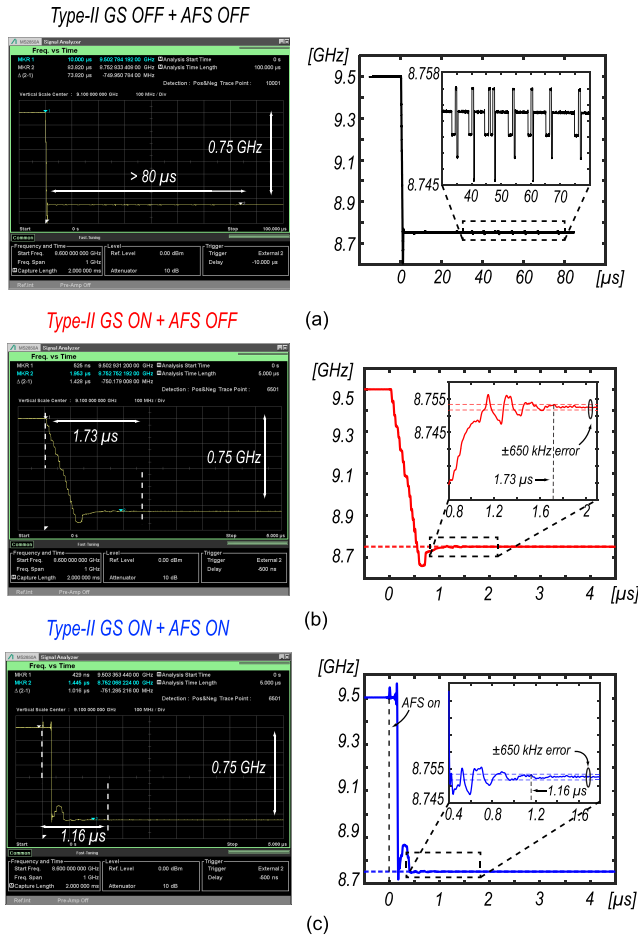


Fig. 17. Measured PLL frequency transients for a negative frequency jump of 0.75 GHz. (a) Settling time longer than  $80\mu\text{s}$  is obtained by turning off all the proposed techniques. (b) GS-II unit reduces the locking time to  $1.73\mu\text{s}$ . (c) Settling performance with both AFS and GS-II units on. The locking time within  $\pm 650\text{ kHz}$ , corresponding to less than 80 ppm, is further reduced to  $1.16\mu\text{s}$ .

to be exactly equivalent to the one discussed in Section III-D, while having the advantage of avoiding the increase of the required DTC range. To improve PLL jitter and fractional spurs performance in fractional- $N$  mode, the DTC range reduction technique in [4] and [14] was adopted.

Fig. 16 shows a die micrograph of the implemented PLL, fabricated in a 28-nm bulk CMOS process. The PLL generates frequencies in the range from 8.5 to 10 GHz, with a power consumption of 20 mW, excluding the input and output buffers, and an active area of  $0.23\text{ mm}^2$ . The input PLL clock is provided by an off-chip SAW oscillator, running at 250 MHz.

Fig. 17 shows the measured PLL frequency transients for a negative frequency jump of 0.75 GHz. When the proposed techniques are turned off, a long and visible limit cycle bounds the locking time to be longer than  $80\mu\text{s}$ . When the type-II gear-shift unit is turned on, the limit cycle is suppressed and the locking time measured within a 650 kHz band error (corresponding to less than 80 ppm<sup>19</sup>), is reduced to  $1.73\mu\text{s}$ . When also the AFS unit is turned on, the locking time is further reduced to  $1.16\mu\text{s}$ , thanks to the estimate of the DCO

<sup>19</sup>This accuracy value was selected to perform a fair comparison with most of the recent works in literature, reported in the table of Fig. 23.

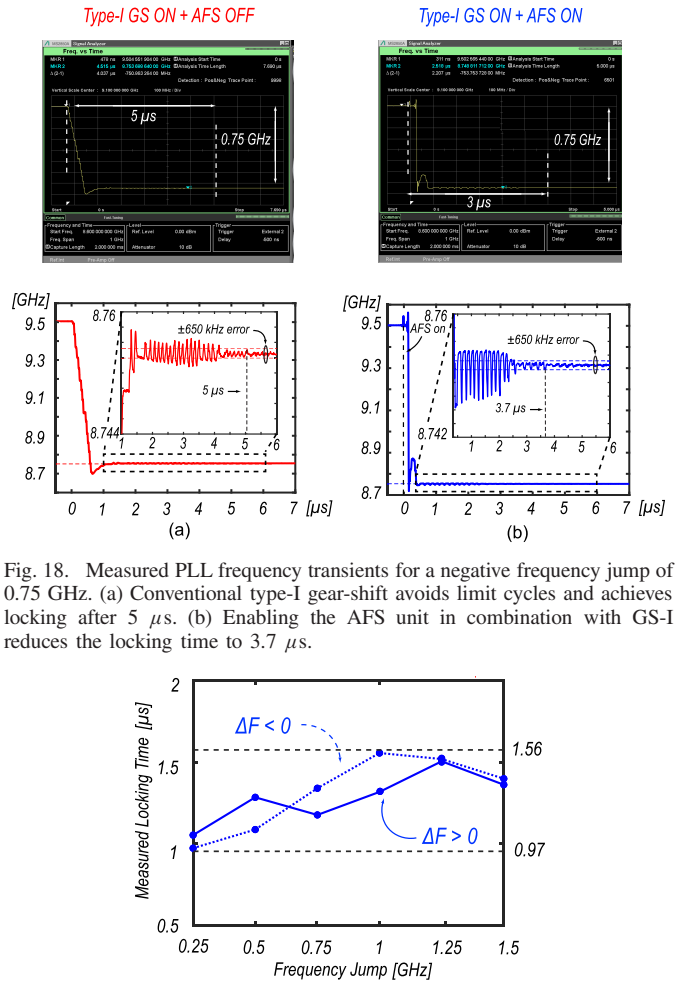


Fig. 18. Measured PLL frequency transients for a negative frequency jump of 0.75 GHz. (a) Conventional type-I gear-shift avoids limit cycles and achieves locking after  $5\mu\text{s}$ . (b) Enabling the AFS unit in combination with GS-I reduces the locking time to  $3.7\mu\text{s}$ .

Fig. 19. PLL locking time performance measured for positive and negative frequency jumps up to 1.5 GHz within the tuning range. The locking time at 80 ppm accuracy always remains below  $1.56\mu\text{s}$ .

frequency gain. The AFS estimation of the DCO gain takes around 150 ns, which corresponds to about 37 clock cycles. The rest of the locking time is given by the residual transient of the auxiliary path and the GS-II unit.

For debug purposes, the implemented GS-II unit can be configured to operate as a conventional GS-I. Fig. 18(a) shows the measured PLL frequency transient when the GS-I is enabled and the AFS unit is turned off. Despite limit cycles are suppressed thanks to the GS-I operation, the subsequent transient is slower when compared with the one using GS-II in Fig. 17(b), thus demonstrating the effectiveness of the proposed technique. A similar result can be derived by comparing the measured frequency transient of Fig. 18(b), where the AFS block was turned on in combination with the GS-I operation, with the one of Fig. 17(c). Fig. 19 shows the PLL locking time performance measured for positive and negative frequency jumps up to 1.5 GHz within the tuning range. The locking time remains always below  $1.56\mu\text{s}$ , which is equivalent to 390 reference cycles.<sup>20</sup>

<sup>20</sup>Based on Fig. 14(a), where the simulated AFS locking time is estimated to be about 200 cycles at large frequency jumps, the proposed AFS and GS-II techniques are expected to equally contribute to the measured 390 locking cycles in this design.

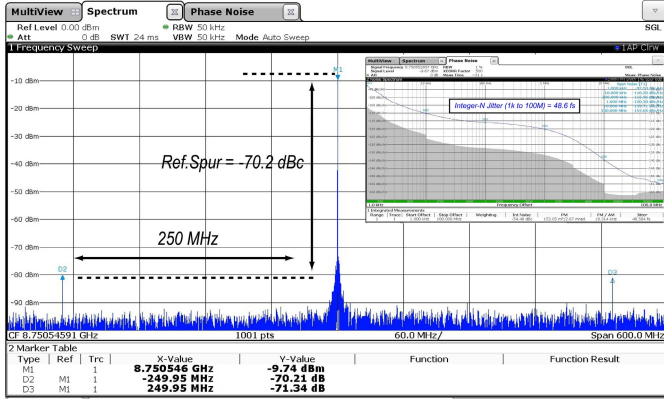


Fig. 20. Measured output spectrum for integer- $N$  operation at the 8.75 GHz frequency channel. The measured integrated jitter, from 1 kHz to 100 MHz, is 48.6 fs, while the measured reference spur is  $-70.2$  dBc at an offset frequency of 250 MHz.

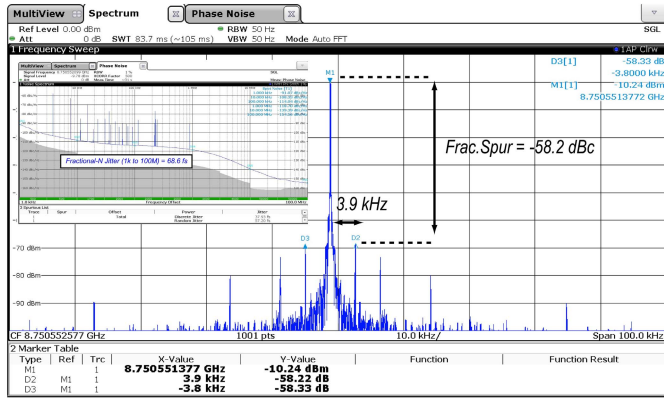


Fig. 21. Measured output spectrum in fractional- $N$  mode, with an offset frequency of 3.9 kHz from the 8.75 GHz channel. The integrated jitter rises to 68.6 fs and the worst case fractional spur is  $-58.2$  dBc at an offset frequency of 3.9 kHz.

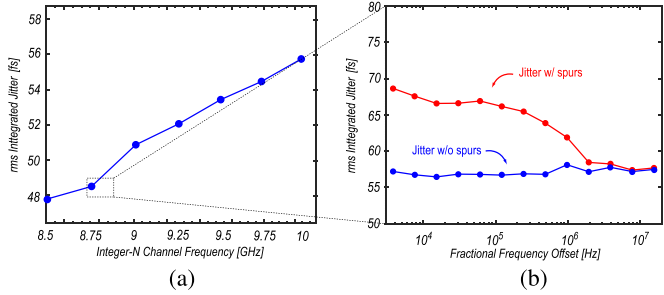


Fig. 22. Measured integrated rms jitter for (a) integer- $N$  operation and (b) fractional- $N$  operation around the 8.75 GHz channel for different frequency offsets.

Fig. 20 shows the integer- $N$  phase noise spectrum at the 8.75 GHz frequency channel. The measured integrated jitter, from 1 kHz to 100 MHz, is 48.6 fs, while the measured reference spur is  $-70.2$  dBc at an offset frequency of 250 MHz. When fractional- $N$  mode is turned on, with an offset frequency of 3.9 kHz from the 8.75 GHz channel (see Fig. 21), the integrated jitter rises to only 68.6 fs while the worst case measured fractional spur, at the same offset frequency, is  $-58.2$  dBc. Fig. 22 shows the measured integrated jitter at different integer- $N$  channels within the tuning range [Fig. 22(a)] and for different fractional frequency offsets from the 8.75 GHz integer- $N$  channel, with and without including the contribution of fractional spurs [Fig. 22(b)].

	This Work	A. Santuccioli [11] ISSCC '20	L. Bertulesi [7] ISSCC '18	C. Tsai [13] JSSC '20	F. ur Rahman [18] JSSC '19	M.-S. Yuan [19] ISSCC '18
PLL Architecture	BBPLL	BBPLL	BBPLL	BBPLL	ADPLL	ADPLL
Locking Method	Aux. BBPDs + Type-II GS + AFS	Aux. BBPD DFER	Aux. BBPDs	Aux. BBPD + GS	Computational Locking	Two Point FM
Type	Fractional-N	Fractional-N	Fractional-N	Integer-N	Integer-N	Fractional-N
Output Frequency [GHz]	8.5-10	12.8-15.2	3.7-4.1	22.5-27.7	1-2	2.1-2.5
Reference Frequency [MHz]	250	500	52	216	50	0.032
Frequency Jump [GHz]	0.25-1.5	1	0.364	0.864	< 1	0.02
Locking Time [ $\mu$ s]	< 1.56	18.55	115	115	< 0.7	< 0.1
Locking Cycles [# Ref. cycles]	< 390	9275	5980	24840	< 35	< 1
Settling Accuracy [ppm]	< 80	70	90	3	N/A	N/A
Integer-N Jitter [fs]	48.6	59	N/A	220	3090	N/A
Frac-N Jitter w/o spurs [fs]	57.2	66.2	183	N/A	N/A	1390
Frac-N Jitter w/ spurs [fs]	68.6	N/A	N/A	N/A	N/A	N/A
Integration Bandwidth [Hz]	1k-100M	1k-100M	1k-30M	10k-20M	N/A	100k-1G
Power Dissipation [mW]	20	19.8	5.28	25	10.8	0.923
FoM <sub>1</sub> [dB]	-251.8	-250.6	-247.5	N/A	N/A	-237.5
FoM <sub>5</sub> [dB]	-250.3	N/A	N/A	N/A	N/A	N/A
Fractional Spur [dBc]	-58.2	-61	-50	N/A	N/A	N/A
Reference Spur [dBc]	-70.2	-80.1	N/A	-65	N/A	N/A
Active Area [ $\mu$ m <sup>2</sup> ]	0.23	0.17	0.61	0.09	0.044	0.24
CMOS Process [nm]	28	28	65	28	65	16

$$^*FoM_1 = 10 \log_{10} [(Power/1mW) / (Frac-N Jitter_{w/o spurs} / 1s)^2] \quad ^**FoM_5 = 10 \log_{10} [(Power/1mW) / (Frac-N Jitter_{w/spurs} / 1s)^2]$$

Fig. 23. Performance comparison with prior art fast-locking DPLLs.

The table in Fig. 23 shows a performance comparison with fast-locking DPLLs. When compared with BBPLLs, this work achieves the best locking time in terms of reference cycles, which is more than an order of magnitude lower than in other published results, while achieving the largest frequency jump and the best integrated jitter, demonstrating that locking time reduction is achieved without impairing rms jitter. When compared with all-DPLLs (ADPLLs), this work reduces the performance gap which is currently existing between BBPLLs and ADPLLs.

## V. CONCLUSION

The work presents an 8.5–10 GHz fractional- $N$  BBPLL implemented in a 28-nm bulk CMOS technology. The system achieves 1.56  $\mu$ s locking time with 68.6 fs integrated jitter by exploiting two novel techniques: 1) a type-II gear-shifting technique to avoid limit cycles and speed-up the locking transient of the main PLL loop and 2) a low-complexity AFS technique to reduce the PLL frequency error upon channel switching using the already existing hardware. The results show an improvement of the locking time of BBPLLs by more than one order of magnitude without impairing the rms-jitter.

## APPENDIX A

Fig. 24 shows a more accurate plot of the PLL state variables during gear-shift. When  $\beta[k]$  is scaled down, a PLL frequency perturbation equal to

$$\Delta F_{p,n} = (\beta_{n-1} - \beta_n) k_f e[k] \quad (12)$$

is instantaneously induced, where  $\beta_n$  and  $\beta_{n-1}$  are the values of  $\beta[k]$  at the new and previous gear-shift steps and  $e[k]$  is the value of the main BBPD signal at the scaling instant.<sup>21</sup> Since the gear-shift scaling occurs after the sign inversion of  $\Delta t[k]$  and  $e[k]$  within each transient section,  $\Delta F_{p,n}$  is opposite with respect to the BBPLL frequency correction term at the end of the previous section, i.e.,  $\pm 2\beta_{n-1} k_f$ . The result is that the PLL capability of removing the initial frequency error  $\Delta F$

<sup>21</sup>The perturbation induced by  $\alpha[k]$  scaling is neglected, as  $\alpha[k] \ll \beta[k]$ .

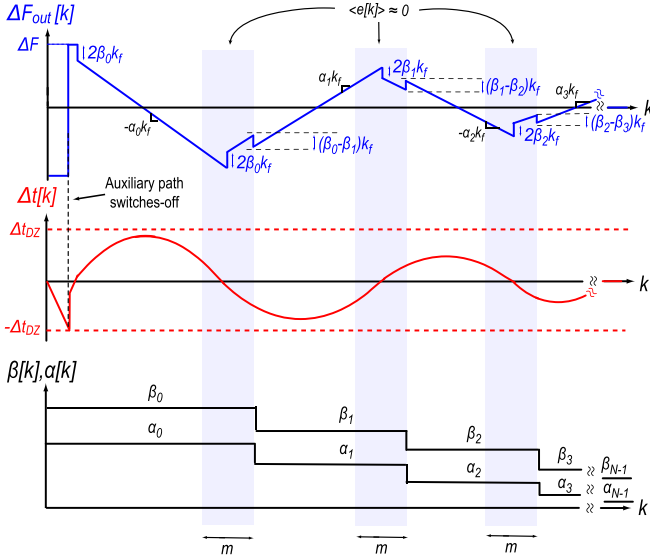


Fig. 24. Transient waveforms of the PLL frequency error  $\Delta F_{out}[k]$ , the PLL time error  $\Delta t[k]$ , and loop-filter gains  $\beta[k]$  and  $\alpha[k]$  during the gear-shift and taking into account the frequency perturbations induced by the loop filter coefficients scaling.

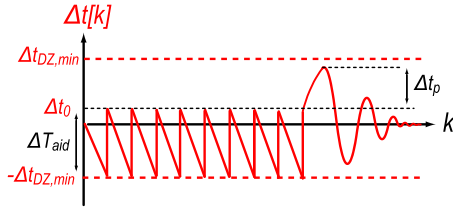


Fig. 25. Transient of the PLL time error  $\Delta t[k]$  for the minimum dead-zone PVT corner and  $\Delta T_{aid} > \Delta t_{dz, min}$ .

is reduced, and the corresponding value of  $\Delta F_{max}$  can be derived as

$$\Delta F_{max} = 2\beta_{gs}k_f \frac{2q-1}{q-1} - \sum_{n=1}^N |\Delta F_{p,n}| \approx \beta_{gs}k_f \frac{3q-1}{q-1} \quad (13)$$

where the sum of all terms  $|\Delta F_{p,n}|$  was subtracted from (6), and  $|\cdot|$  is the absolute value operation.

#### APPENDIX B

Fig. 25 depicts the operation of the auxiliary path in Section II for the minimum dead-zone PVT corner, i.e.,  $\Delta t_{dz} = \Delta t_{dz, min}$ . If  $\Delta T_{aid} > \Delta t_{dz, min}$ , the initial time error  $\Delta t_0$  after the auxiliary path switch-off would be positive. As a consequence the maximum value of the time error overshoot  $\Delta t_p$  to avoid triggering limit cycles, i.e.,  $\Delta t_{p, max}$ , would be smaller than the dead-zone, and  $\Delta F_{crit}$  in such condition can be found by substituting the term  $\Delta t_{dz}$  in (3) with  $\Delta t_{p, max} = \Delta t_{dz, min} - \Delta t_0$ . Since  $\Delta F_{crit} \propto (\Delta t_{p, max})^{1/2}$ , the minimum dead-zone PVT corner is the worst case, and therefore, to achieve a robust operation,  $\Delta t_{p, max}$  should be maximized in this condition. To do so, the value of  $\Delta T_{aid}$  was chosen to match  $\Delta t_{dz, min}$ .<sup>22</sup>

<sup>22</sup>The value of  $\Delta T_{aid}$  varies across the tuning range, and, from (1), it is maximized for the minimum  $F_{out}$ . Therefore,  $\Delta T_{aid, max} = \Delta t_{dz, min}$  was enforced, since this is the worst case where the largest positive  $\Delta t_0$  takes place during the transient.

Note that, for what concerns limit cycles, only the fine BBPD operation in Fig. 15 must be taken into account, as the coarse BBPD eventually switches off near the end of the transient. The minimum value of the implemented fine BBPD dead-zone is about  $\Delta t_{dz, min} = 140$  ps from circuit-level simulations on PVT corners, resulting in a worst case  $\Delta F_{crit} \approx 14$  MHz, which is close to its nominal value of about 16 MHz and therefore not being a problem in our design.

#### REFERENCES

- [1] W. Wu *et al.*, "A 28-nm 75- $f_{rms}$  analog fractional-N sampling PLL with a highly linear DTC incorporating background DTC gain calibration and reference clock duty cycle correction," *IEEE J. Solid-State Circuits*, vol. 54, no. 5, pp. 1254–1265, May 2019.
- [2] D. Turker *et al.*, "A 7.4-to-14 GHz PLL with 54  $f_{rms}$  jitter in 16 nm FinFET for integrated RF-data-converter SoCs," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 378–380.
- [3] M. Mercandelli *et al.*, "A 12.5-GHz fractional-N type-I sampling PLL achieving 58-fs integrated jitter," *IEEE J. Solid-State Circuits*, vol. 57, no. 2, pp. 505–517, Feb. 2022.
- [4] W. Wu *et al.*, "A 14-nm ultra-low jitter fractional-N PLL using a DTC range reduction technique and a reconfigurable dual-core VCO," *IEEE J. Solid-State Circuits*, vol. 56, no. 12, pp. 3756–3767, Dec. 2021.
- [5] D.-G. Lee and P. P. Mercier, "A sub-mW 2.4-GHz active-mixer-adopted sub-sampling PLL achieving an FoM of  $-256$  dB," *IEEE J. Solid-State Circuits*, vol. 55, no. 6, pp. 1542–1552, Jun. 2020.
- [6] A. Sharkia, S. Mirabbasi, and S. Shekhar, "A 0.01  $\text{mm}^2$  4.6-to-5.6 GHz sub-sampling type-I frequency synthesizer with  $-254$  dB FOM," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 256–258.
- [7] L. Bertulesi, L. Grimaldi, D. Cherniak, C. Samori, and S. Levantino, "A low-phase-noise digital bang-bang PLL with fast lock over a wide lock range," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 252–254.
- [8] D. Cherniak *et al.*, "A 15.6–18.2 GHz digital bang-bang PLL with  $-63$  dBc in-band fractional spur," in *Proc. IEEE Radio Freq. Integr. Circuits Symp. (RFIC)*, Jun. 2018, pp. 36–39.
- [9] S. M. Dartizio *et al.*, "A 68.6  $f_{rms}$ -total-integrated-jitter and 1.56  $\mu\text{s}$ -locking-time fractional-N bang-bang PLL based on type-II gear shifting and adaptive frequency switching," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 1–3.
- [10] S. M. Dartizio *et al.*, "A 12.9-to-15.1-GHz digital PLL based on a bang-bang phase detector with adaptively optimized noise shaping," *IEEE J. Solid-State Circuits*, vol. 57, no. 6, pp. 1723–1735, Jun. 2022.
- [11] A. Santiccioli *et al.*, "A 66- $f_{rms}$  jitter 12.8-to-15.2-GHz fractional-N bang-bang PLL with digital frequency-error recovery for fast locking," *IEEE J. Solid-State Circuits*, vol. 55, no. 12, pp. 3349–3361, Dec. 2020.
- [12] A. Santiccioli *et al.*, "A 98.4 fs-jitter 12.9-to-15.1 GHz PLL-based LO phase-shifting system with digital background phase-offset correction for integrated phased arrays," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 456–458.
- [13] C.-H. Tsai, Z. Zong, F. Pepe, G. Mangraviti, J. Craninckx, and P. Wambacq, "Analysis of a 28-nm CMOS fast-lock bang-bang digital PLL with 220-fs RMS jitter for millimeter-wave communication," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1854–1863, Jul. 2020.
- [14] D. Tasca, M. Zanuso, G. Marzin, S. Levantino, C. Samori, and A. L. Lacaita, "A 2.9–4.0-GHz fractional-N digital PLL with bang-bang phase detector and 560- $f_{rms}$  integrated jitter at 4.5-mW power," *IEEE J. Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec. 2011.
- [15] L. Bertulesi *et al.*, "A 30-GHz digital sub-sampling fractional-N PLL with  $-238.6$ -dB jitter-power figure of merit in 65-nm LP CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 3493–3502, Dec. 2019.
- [16] T. Seong, Y. Lee, S. Yoo, and J. Choi, "A 320- $f_{rms}$  jitter and  $-75$ -dBc reference-spur ring-DCO-based digital PLL using an optimal-threshold TDC," *IEEE J. Solid-State Circuits*, vol. 54, no. 9, pp. 2501–2512, Dec. 2019.
- [17] J. Kim *et al.*, "An ultra-low-jitter, mmW-band frequency synthesizer based on digital subsampling PLL using optimally spaced voltage comparators," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 3466–3477, Dec. 2019.
- [18] F. U. Rahman, G. Taylor, and V. Sathe, "A 1–2 GHz computational-locking ADPLL with sub-20-cycle locktime across PVT variation," *IEEE J. Solid-State Circuits*, vol. 54, no. 9, pp. 2487–2500, Sep. 2019.

- [19] C.-C. Li, M.-S. Yuan, C.-C. Liao, Y.-T. Lin, C.-H. Chang, and R. B. Staszewski, "All-digital PLL for Bluetooth low energy using 32.768-kHz reference clock and  $\leq 0.45$ -V supply," *IEEE J. Solid-State Circuits*, vol. 53, no. 12, pp. 3660–3671, Dec. 2018.
- [20] C.-C. Hung and S.-I. Liu, "A 40-GHz fast-locked all-digital phase-locked loop using a modified bang-bang algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 6, pp. 321–325, Jun. 2011.
- [21] Q. Huang, C. Zhan, and J. Burm, "A low-complexity fast-locking digital PLL with multi-output bang-bang phase detector," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Oct. 2016, pp. 418–420.
- [22] J.-M. Lin and C.-Y. Yang, "A fast-locking all-digital phase-locked loop with dynamic loop bandwidth adjustment," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 10, pp. 2411–2422, Oct. 2004.
- [23] R. Nonis, W. Grollitsch, T. Santa, D. Cherniak, and N. D. Dalt, "DigPLL-lite: A low-complexity, low-jitter fractional-N digital PLL architecture," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3134–3145, Dec. 2013.
- [24] C.-T. Wu, W.-C. Shen, W. Wang, and A.-Y. Wu, "A two-cycle lock-in time ADPLL design based on a frequency estimation algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 6, pp. 430–434, Jun. 2010.
- [25] D.-S. Kim, H. Song, T. Kim, S. Kim, and D.-K. Jeong, "A 0.3–1.4 GHz all-digital fractional-N PLL with adaptive loop gain controller," *IEEE J. Solid-State Circuits*, vol. 45, no. 11, pp. 2300–2311, Nov. 2010.
- [26] L. Bertulesi, D. Cherniak, M. Mercandelli, C. Samori, A. L. Lacaíta, and S. Levantino, "Novel feed-forward technique for digital bang-bang PLL to achieve fast lock and low phase noise," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 5, pp. 1858–1870, May 2022.
- [27] L. Avallone, M. Mercandelli, A. Santiccioli, M. P. Kennedy, S. Levantino, and C. Samori, "A comprehensive phase noise analysis of bang-bang digital PLLs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 7, pp. 2775–2786, Jul. 2021.
- [28] T.-K. Kuan and S.-I. Liu, "A bang bang phase-locked loop using automatic loop gain control and loop latency reduction techniques," *IEEE J. Solid-State Circuits*, vol. 51, no. 4, pp. 821–831, Apr. 2016.
- [29] A. Zanchi, A. Bonfanti, S. Levantino, and C. Samori, "General SSCR vs. Cycle-to-cycle jitter relationship with application to the phase noise in PLL," in *Proc. Southwest Symp. Mixed-Signal Design*, 2001, pp. 32–37.
- [30] G. Marucci, S. Levantino, P. Maffezzoni, and C. Samori, "Analysis and design of low-jitter digital bang-bang phase-locked loops," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 26–36, Jan. 2014.
- [31] R. B. Staszewski and P. T. Balsara, "All-digital PLL with ultra fast settling," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 2, pp. 181–185, Feb. 2007.
- [32] D. Cherniak, L. Grimaldi, L. Bertulesi, R. Nonis, C. Samori, and S. Levantino, "A 23-GHz low-phase-noise digital bang-bang PLL for fast triangular and sawtooth chirp modulation," *IEEE J. Solid-State Circuits*, vol. 53, no. 12, pp. 3565–3575, Dec. 2018.
- [33] P. T. Renukaswamy, N. Markulic, P. Wambacq, and J. Craninckx, "A 12-mW 10-GHz FMCW PLL based on an integrating DAC with 28-kHz RMS-frequency-error for 23-MHz/ $\mu$ s slope and 1.2-GHz chirp-bandwidth," *IEEE J. Solid-State Circuits*, vol. 55, no. 12, pp. 3294–3307, Dec. 2020.