A Low-Spur Fractional-N PLL Based on a Time-Mode Arithmetic Unit

Gao, Zhong; He, Jingchu; Fritz, Martin; Shen, Yiyu; Zong, Zhirui; Spalink, Gerd; Alavi, Morteza S.; Staszewski, Robert Bogdan; Babaie, Masoud; More Authors

DOI
10.1109/JSSC.2022.3209338

Publication date
2023

Document Version
Final published version

Published in
IEEE Journal of Solid-State Circuits

Citation (APA)
Gao, Z., He, J., Fritz, M., Shen, Y., Zong, Z., Spalink, G., Alavi, M. S., Staszewski, R. B., Babaie, M., & More Authors (2023). A Low-Spur Fractional-N PLL Based on a Time-Mode Arithmetic Unit. *IEEE Journal of Solid-State Circuits*, *58*(6), 1552-1571. https://doi.org/10.1109/JSSC.2022.3209338

A Low-Spur Fractional-N PLL Based on a Time-Mode Arithmetic Unit

Zhong Gao, *Graduate Student Member, IEEE*, Jingchu He, Martin Fritz, Jiang Gong, *Student Member, IEEE*, Yiyu Shen, *Member, IEEE*, Zhirui Zong, *Member, IEEE*, Peng Chen, *Member, IEEE*, Gerd Spalink, Ben Eitel, Morteza S. Alavi, *Member, IEEE*, Robert Bogdan Staszewski, *Fellow, IEEE*, and Masoud Babaie, *Senior Member, IEEE*

***Abstract*—This article introduces a low-jitter low-spur fractional-N phase-locked loop (PLL) adopting a new concept of a time-mode arithmetic unit (TAU) for phase error extraction. The TAU is a time-signal processor that calculates the weighted sum of input time offsets. It processes two inputs—the period of a digitally controlled oscillator (DCO) and the instantaneous time offset between the DCO and reference clock edges—and then extracts the DCO phase error by calculating their weighted sum. The prototype, implemented in 40-nm CMOS, achieves 182-fs rms jitter with 3.5-mW power consumption. In a near-integer channel, it shows the worst fractional spur below −59 dBc. Under considerable supply or temperature variations, the worst spur still remains below −51.7 dBc without any background calibration tracking.**

***Index Terms*—Digital-to-time converter (DTC), fractional spur, phase-locked loop (PLL), process voltage and temperature (PVT), time-mode arithmetic unit (TAU).**

I. Introduction

THE sub-sampling technique [1], [2], [3], [4], [5] and, more specifically, the narrow-range phase-detection concept [6], [7] have contributed to a significant improvement of phase noise (PN) in phase-locked loops (PLLs). Such techniques ensure that the phase detector (PD) can expect a near-zero input so that its gain can be enlarged to suppress the noise contribution of subsequent loop blocks.

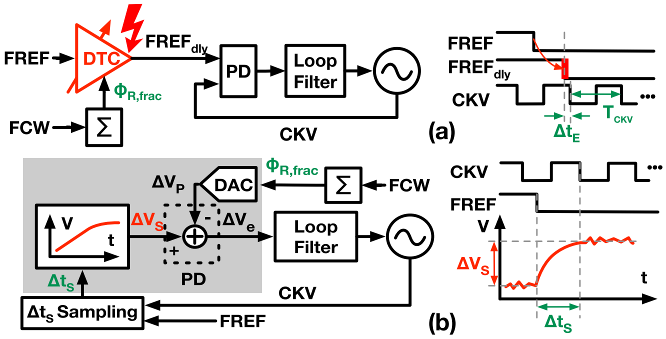


Fig. 1. Time-offset cancellation strategies to narrow the required input range of PD using (a) DTC and (b) voltage-domain cancellation.

In applying the narrow-range phase-detection concept to fractional-N PLLs, digital-to-time converters (DTCs) are widely explored [8], [9], [10], [11], [12], [13]. Fig. 1(a) illustrates a conceptual PLL (similar to that in [14]) relying on a DTC to cancel the instantaneous time offset from the significant (here, falling) edge of the reference clock (FREF) to that of the variable oscillator clock (CKV). This time offset spans from zero to the CKV period T_{CKV} (i.e., "time base") and is predicted by $\phi_{\mathrm{R,frac}} \in [0, 1)$, i.e., the fractional part of the accumulated frequency control word (FCW) [15]. According to this prediction, the DTC launches a delayed FREF, FREF$_{\mathrm{dly}}$, which is substantially aligned with the relevant CKV edge in order to narrow down the input range of the PD.

This DTC-based solution is highly effective in improving PN. However, it potentially introduces high fractional spurs at the PLL output since the DTC delay can easily depart from its nominal expected value of $(1-\phi_{\mathrm{R,frac}}) \cdot T_{\mathrm{CKV}}$. Such a mismatch stems from the underlying principle of DTCs—delaying input edges based on the circuit's nominal intrinsic latency, e.g., the propagation delay of the elements in a delay-chain-based DTC [16]. This is markedly distinct from the conventional digital-to-analog converters (DACs), which generate signals by scaling a stable and accurate base, e.g., a bandgap reference voltage. Given the sensitivity of the circuit's intrinsic latency to process, voltage, and temperature (PVT) variations [17], [18], [19], an extra effort is required in tracking and protecting the DTC's transfer function (i.e., from $\phi_{\mathrm{R,frac}}$ to DTC delay), so as to prevent the associated PLL spurs from arising. For example,

Manuscript received 10 June 2022; revised 18 August 2022; accepted 12 September 2022. Date of publication 13 October 2022; date of current version 26 May 2023. This article was approved by Associate Editor Kenichi Okada. This work was supported by Sony Semiconductor Solutions. *(Corresponding authors: Zhong Gao; Robert Bogdan Staszewski.)*

Zhong Gao, Yiyu Shen, Morteza S. Alavi, and Masoud Babaie are with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: z.gao-2@tudelft.nl).

Jingchu He was with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands. She is now with NXP Semiconductor, 6534 AE Eindhoven, The Netherlands.

Martin Fritz, Gerd Spalink, and Ben Eitel are with the Stuttgart Technology Center, Sony Europe B.V., 70327 Stuttgart, Germany.

Jiang Gong is with the Department of Quantum and Computer Engineering, Delft University of Technology, 2628 CJ Delft, The Netherlands.

Zhirui Zong was with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands. He is now with Microelectronics Thrust, The Hong Kong University of Science and Technology, Guangzhou 511400, China.

Peng Chen was with the School of Electrical and Electronic Engineering, University College Dublin, Dublin 4, Ireland. He is now with Wuxi Grand-Micro, Wuxi 214063, China.

Robert Bogdan Staszewski is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin 4, Ireland, and also with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: robert.staszewski@ucd.ie).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2022.3209338.

Digital Object Identifier 10.1109/JSSC.2022.3209338

Tasca *et al.* [8], Liu *et al.* [20], and Un *et al.* [21] track the drift of the DTC transfer gain in the background. Wu *et al.* [7] and Markulic *et al.* [22] protect the DTC delay from supply variations with dedicated low-drop regulators (LDOs), so as to alleviate any memory effects in DTC's transfer function. Markulic *et al.* [22] further use a complementary dummy DTC to reduce the time-varying supply perturbations resulting from the main DTC. These countermeasures, however, only exhibit limited capabilities in suppressing the DTC-related spurs. When an extremely low spurious level is desired, the DTC codes might need to be modulated to smear the spurs into the noise floor [22], [23], [24]. These extra efforts complicate the design of the overall PLL system and degrade its power efficiency.

Instead of relying on the circuit propagation delay, Wu *et al.* [17], Chen *et al.* [19], and Liao and Dai [25] cancel the instantaneous fractional-N time offset in the voltage domain. A conceptual example emulating [17] is presented in Fig. 1(b). The time offset between FREF and its subsequent CKV edge, Δt_{S}, is converted into voltage $\Delta \mathrm{V}_{\mathrm{S}}$ by the charging curve characteristic. The PLL cancels $\Delta \mathrm{V}_{\mathrm{S}}$ with its prediction ($\Delta \mathrm{V}_{\mathrm{P}}$) to extract the phase error information in the voltage domain ($\Delta \mathrm{V}_{\mathrm{e}}$). Accurate error extraction here requires a charging curve of constant slope since the voltage prediction assumes a linear time-to-voltage conversion. Such a dependence is also imperfect because the slope is generated by (dis)charging a capacitor through a current source, which raises two issues: 1) it requires a stable current reference, which is costly and 2) it invokes a tradeoff issue between noise, linearity, and power: linearity of the (dis)charging slope is degraded by the finite impedance of the current source (i.e., an MOS transistor). Therefore, circuit-level techniques, such as cascoding [12], [26], seem mandatory. Nevertheless, they consume significant voltage headroom, thereby degrading the noise. To combat the noise, the associated node capacitance and current must be enlarged. However, a larger current implies not only higher power but also a wider current source transistor, which lowers the impedance and possibly launches a new round of tradeoffs.

The dilemmas of these two discussed methods root in their dependence on the PVT-sensitive physical parameters, i.e., the intrinsic circuit latency of the DTC and the (dis)charging slope in the voltage-domain cancellation. Mathematically, a "golden" base for the fractional-N time-offset cancellation is T_{CKV} since the time offset is predicted by $(1-\phi_{\mathrm{R,frac}})\cdot T_{\mathrm{CKV}}$. In terms of implementation, T_{CKV} is also accurate and stable since it is intrinsically tracked by the PLL. Therefore, we propose a new time-offset cancellation method adopting T_{CKV} as its base, which can be considered analogous to the aforementioned reference voltage in a DAC.[1] The proposed method employs a time-mode arithmetic unit (TAU) processor [29] that takes timestamp offsets as inputs and outputs their weighted sum, also in the time domain. Within each PLL cycle, TAU takes both the timestamps defining T_{CKV}, as well as the timestamps defining Δt_{S}, i.e., the offset between the oscillator and reference clock edges. Then, the weighted sum of their offsets is calculated to extract the desired information (i.e., time error Δt_{E} input to the PD). With such a "golden" time base, the TAU-based method can exhibit high linearity and built-in resilience to supply and temperature variations. This simplifies the overall PLL system design and helps to suppress the generated spurs. As an extra bonus, TAU can advantageously amplify the desired time residue, thereby suppressing the noise contributions from subsequent loop blocks.

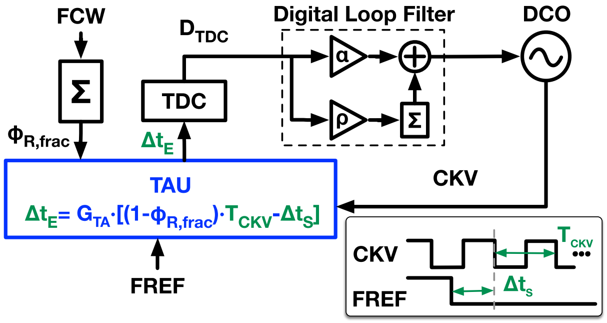


Fig. 2. Conceptual diagram of the proposed TAU-based PLL.

II. Principle of the Proposed PLL

A. Conceptual Architecture

Fig. 2 shows a conceptual diagram of the proposed fractional-N PLL. To track the reference phase by the digitally controlled oscillator (DCO), the proposed TAU extracts the time error (Δt_{E}) between the FREF and CKV timestamps. This Δt_{E} is quantized by the time-to-digital converter (TDC) and input to the digital loop filter for the DCO phase error correction.

Generally, Δt_{E} "hides" within Δt_{S}, which is the instantaneous "raw" time offset between FREF and the first subsequent CKV falling edge, with theoretical prediction of $(1-\phi_{\mathrm{R,frac}})\cdot T_{\mathrm{CKV}}$. Therefore, extracting Δt_{E} requires canceling Δt_{S} with its prediction. In the proposed system, the TAU samples Δt_{S} and T_{CKV}, and then calculates their weighted sum to extract Δt_{E}. To further help with suppressing the TDC quantization noise, the TAU also time-amplifies the extracted error by G_{TA} before feeding it to the TDC. Thus, the TAU's output can be described as

$$\Delta t_{\mathrm{E}} = G_{\mathrm{TA}}\cdot\left[\left(1-\phi_{\mathrm{R,frac}}\right)\cdot T_{\mathrm{CKV}} - \Delta t_{\mathrm{S}}\right]. \tag{1}$$

More abstractly, if T_{CKV} and Δt_{S} are viewed as general inputs, and G_{TA} and $\phi_{\mathrm{R,frac}}$ are treated as their weights, the TAU's function can be generalized as producing the weighted sum of its inputs

$$\Delta t_{\mathrm{out}} = \sum_{i=1}^{n} w_i\cdot\Delta t_i \tag{2}$$

[1] It is interesting to note that Narayanan *et al.* [27] phase-interpolate new edges from a quadrature RF clock source to substantially cancel the time offset, just like DTC does. That method can be also regarded as utilizing a "golden" time base of $T_{\mathrm{CKV}}/4$. However, cascading many stages of phase interpolators to achieve fine resolution can be quite bulky and power-hungry. A similar approach in [28] uses eight internal phases from an RF oscillator that is followed by a ÷4 divider to reduce the time offset by a factor of 8. The quantization is coarser (i.e., 3 bits to yield a $T_{\mathrm{CKV}}/8$ "golden" time base), but the analog interpolators are avoided.

where Δt_i is the ith input time offset, w_i is the weight applied to Δt_i, n is the total number of inputs, and Δt_{out} is the output time offset. Note that Δt_i and Δt_{out} are generally defined as the time offsets between arbitrary edges.

To realize this conceptual PLL system, we first realize this generalized TAU and then program it to calculate the result required by (1).

B. Evolution From Time Register to TAU

The starting point for implementing TAU is a time register (TR), which takes pulsewidths as inputs, holds them, and then outputs their sum in a complementary form [30]. Fig. 3(a) illustrates how to achieve these functions with a simplified RC model of TR [31]. Before a new execution cycle, capacitor C is charged to an initial voltage V_{init} by closing the charging switch SWC. After SWC is disconnected, the TR processes the active-low pulses on the discharge switch SWD by means of storing their pulsewidths as voltage drops on capacitor C. For example, during the first pulse, the switch SWD is closed to discharge capacitor C through resistor R. After Δt_1, the duration of the first pulse, the voltage on the capacitor V_{C} drops from V_{init} to $V_1 = V_{\text{init}} \cdot \exp(-\Delta t_1/\tau_0)$, where $\tau_0 = R \cdot C$ is the RC time constant for discharging. Hence, the input time Δt_1 is recorded in the TR as a voltage drop $V_{\text{init}} - V_1$. Similarly, after the second pulse, V_{C} drops to $V_2 = V_1 \cdot \exp(-\Delta t_2/\tau_0) = V_{\text{init}} \cdot \exp(-\Delta t_1/\tau_0 - \Delta t_2/\tau_0)$. The new input time Δt_2 is internally summed with the prestored Δt_1 and recorded as $V_{\text{init}} - V_2$. The TR can continue to process more inputs as long as V_{C} is higher than V_{th}, i.e., the threshold voltage of the level-crossing comparator (slicer). Assuming that the TR has processed n pulses in total, the final V_{C} becomes

$$V_n = V_{n-1} \cdot \exp\left(-\frac{\Delta t_n}{\tau_0}\right) = V_{\text{init}} \cdot \exp\left(-\sum_{i=1}^{n} \frac{\Delta t_i}{\tau_0}\right) \quad (3)$$

where Δt_i is the width of the ith pulse. To read the recorded time, SWD is pulled down to discharge the capacitor voltage V_{C} to below V_{th}, thereby asserting the comparator's output CMP. The delay between the last SWD and CMP falling edges reflects the processed result, which is an offset (the duration in which V_{C} is continuously discharged from V_{init} to V_{th}) minus the sum of all time inputs

$$\Delta t_{\text{out}} = \tau_0 \ln \frac{V_n}{V_{\text{th}}} = \tau_0 \ln \frac{V_{\text{init}}}{V_{\text{th}}} - \sum_{i=1}^{n} \Delta t_i. \quad (4)$$

A quick comparison between (4) and (2) suggests a crucial limitation of the TR—its weight for each Δt_i can only be 1 instead of an arbitrary w_i. The weighted TR (WTR) shown in Fig. 3(b) overcomes this limitation by replacing the fixed resistor R and capacitor C with the variable ones, R_V and C_V. With this change, the WTR acquires a new degree of freedom, i.e., the variable RC time constant $\tau = R_V \cdot C_V$, to influence each pulse's discharge speed and the resulting voltage drop on V_{C}.

Accordingly, the WTR's final output becomes

$$\Delta t_{\text{out}} = \tau_{\text{out}} \cdot \ln \frac{V_{\text{init}}}{V_{\text{th}}} - \sum_{i=1}^{n} \frac{\tau_{\text{out}}}{\tau_i} \Delta t_i \quad (5)$$

where τ_i is the RC time constant for Δt_i, and τ_{out} is the RC time constant for the final output discharge. Here, an arbitrary weight, $w_i = \tau_{\text{out}}/\tau_i$, is effectively applied to Δt_i.

Although the WTR achieves the weighted sum $[\sum_{i=1}^{n}(\tau_{\text{out}}/\tau_i) \cdot \Delta t_i]$, the offset term $\tau_{\text{out}} \cdot \ln(V_{\text{init}}/V_{\text{th}})$ in its output raises undesired issues. This term indicates the WTR's sensitivity to voltages, i.e., V_{init} and V_{th}, and physical parameters, e.g., τ_{out}, which can ultimately lead to a severe PVT susceptibility. This term is advantageously canceled in a differential WTR (DWTR) configuration shown in Fig. 3(c). Two identical WTRs operate there in parallel and share the common resistive and capacitive tuning terminals, RT and CT. Hence, the same RC time constant τ_i is applied to their ith input pair (i.e., $\Delta t_{i,\text{P}}$ and $\Delta t_{i,\text{N}}$). Non-shared pins of the two WTRs are distinguished with subscripts P and N. The outputs of two individual WTRs follow the same rule as (5). Combining these outputs differentially, the PVT-sensitive offset terms cancel out each other

$$\Delta t_{\text{out}} = \Delta t_{\text{out,N}} - \Delta t_{\text{out,P}} = \sum_{i=1}^{n} \frac{\tau_{\text{out}}}{\tau_i} \cdot \left(\Delta t_{i,\text{P}} - \Delta t_{i,\text{N}}\right). \quad (6)$$

Nevertheless, the differential inputs and output required by the DWTR are too complex to use—they are the pulsewidth differences ($\Delta t_{i,\text{P}} - \Delta t_{i,\text{N}}$ and $\Delta t_{\text{out,N}} - \Delta t_{\text{out,P}}$), instead of the time differences defined in (2). Therefore, their form is redefined. For the output, we simply impose a constraint that the last falling edges on SWD_{P} and SWD_{N} must be launched simultaneously. Then, the differential output Δt_{out} is reinterpreted as the time offset between CMP_{P} and CMP_{N}, which equals $\Delta t_{\text{out,N}} - \Delta t_{\text{out,P}}$ [see Fig. 3(c)].

For the input form conversion, the proposed TAU employs a phase/frequency detector (PFD). As shown in Fig. 3(d), the PFD bridges the gap between the overall TAU input, i.e., the time difference between TIN_{P} and TIN_{N} falling edges, and the DWTR input, i.e., the width difference of the pulse-pair on SWD_{P} and SWD_{N}. To do so, the PFD first pulls down SWD_{P} and SWD_{N} at the TIN_{P} and TIN_{N} falling edges, respectively. Once both SWDs become low, the PFD resets itself to pull them up simultaneously. By doing so, the PFD converts the input time difference to the pulsewidth difference. However, during the TAU output processing, the SWDs should stay LOW to keep discharging the WTRs until both CMPs' falling edges are asserted. At this moment, the PFD should not revert the SWDs to HIGH because this would disrupt the output process. Therefore, when $\overline{\text{READ}} = 0$ triggers the final output, it also blocks the PFD's reset (the second mode of PFD) and, thus, the SWD recovery.

The output of the proposed TAU is

$$\Delta t_{\text{out}} = \sum_{i=1}^{n} \frac{\tau_{\text{out}}}{\tau_i} \Delta t_i \quad (7)$$

where Δt_i is the input time difference between the ith pair of the $\text{TIN}_{\text{P/N}}$ falling edges, and Δt_{out} is the output time offset between $\text{CMP}_{\text{P/N}}$. Note that Δt_i here can be either positive or negative depending on the corresponding leading edge on the TINs. The TAU calculates the weighted sum of all inputs, whose weights can be manipulated by tuning the

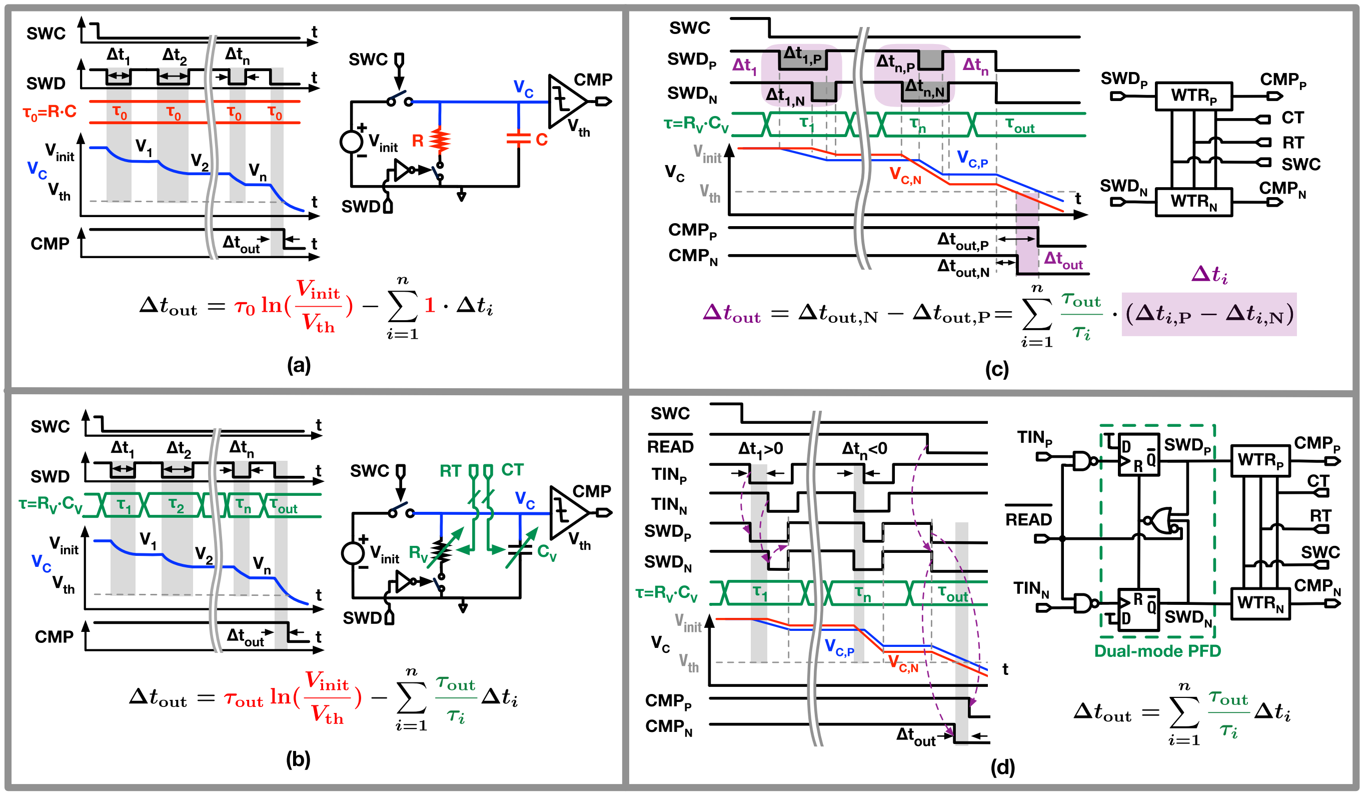


Fig. 3. Conceptual and timing diagrams of (a) TR, (b) WTR, (c) DWTRs, and (d) TAU. Δt_i is the ith time-domain input. Δt_{out} is the time domain output. Detailed definitions of Δt_i and Δt_{out} are circuit-dependent.

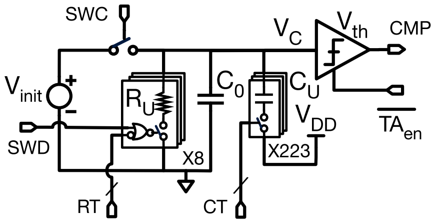


Fig. 4. Schematic of the implemented WTR.

corresponding RC time constants (τ_{out} and τ_i's). Therefore, the TAU's definition in Section II-A can be satisfied. However, one may still question the equivalence between (7) and (2) since the weights are positive-only in the former (τ_{out}/τ_i) but can also be negative in the latter (ω_i). This limitation can be addressed by transferring the weight's ± sign to its associated input Δt_i, whose polarity is determined by the corresponding leading edge on the TINs [see $TIN_{P/N}$ in Fig. 3(d)]. In our implementation shown later, we achieve the negative weight by deliberately swapping the leading-falling edges in the corresponding active-low SWD pulse-pair.

C. RC Tuning in the WTR

To further detail the weight control in (7) by means of τ_{out}/τ_i, Fig. 4 reifies the variable resistance and capacitance introduced in the conceptual WTR of Fig. 3(b). The variable resistor is implemented with a switched-resistor (SR) bank, consisting of parallel unit resistors, R_U. RT determines the number of actively discharging R_U's (8 in total). Meanwhile, the variable capacitor is realized with a fixed capacitor C_0 and a switched-capacitor (SC) bank, consisting of parallel unit capacitors, C_U, whose active count is controlled by CT. Therefore, the RC time constant can be controlled as

$$\tau = \frac{R_U}{\mathrm{RT}} \cdot (C_0 + C_U \cdot \mathrm{CT}). \quad (8)$$

Note that, during the complete TAU execution cycle (from the reset to output), increasing CT would engage new V_{init}-precharged capacitor units, which would lead to charge sharing, thus erroneously increasing the V_C voltage. Therefore, CT is constrained to stay constant or decrease when processing the TAU inputs (see Fig. 5).

The RC tuning of WTR is introduced here to pave the way for the TAU control flow design in Section II-D. Other details are delayed until Section III-E.

D. TAU Control Flow Within the Proposed PLL

The basis of the TAU in the proposed PLL system stems from (1). It was then abstracted as computing the weighted sum of its time inputs, which also generalizes the TAU functionality, i.e., (7). To program the TAU to execute (1), we designed a dedicated control flow to ensure that the TAU receives T_{CKV} and Δt_S [i.e., time inputs of (1)], assigns proper weights to them, and outputs the weighted sum.

According to Fig. 5, the TAU processes four time-domain inputs in a single execution cycle. By tuning the RT and CT

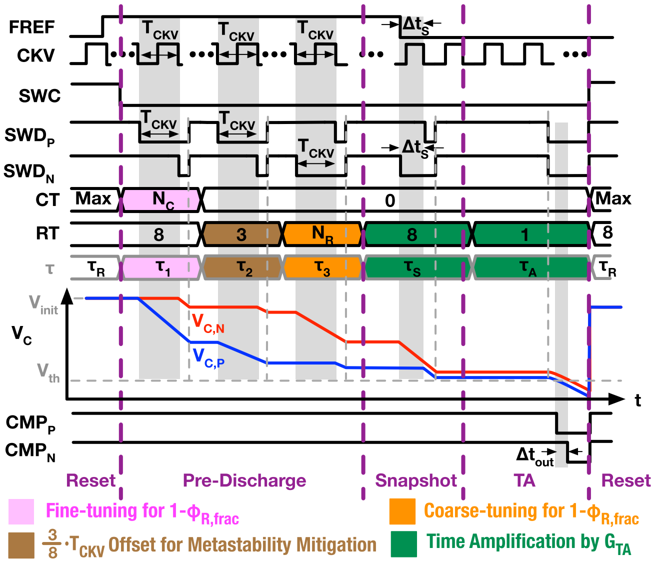


Fig. 5. Timing diagram of the DWTRs' in a complete TAU execution cycle.

control pins, different RC time constants (τ's) can be assigned to each input. According to (7), the resulted output is

$$\Delta t_{\text{out}} = \frac{\tau_A}{\tau_1}T_{\text{CKV}} + \frac{\tau_A}{\tau_2}T_{\text{CKV}} - \frac{\tau_A}{\tau_3}T_{\text{CKV}} - \frac{\tau_A}{\tau_S}\Delta t_S \tag{9}$$

where τ_1, τ_2, and τ_3 are the RC time constants during the first to third discharge, while τ_S and τ_A are those during the Δt_S sampling and final output, respectively. The minus signs result from the swapped leading-falling edges in the corresponding SWD pulse-pairs, as discussed in Section II-B. By replacing the τ symbols with their respective components in (8), Δt_{out} becomes

$$\Delta t_{\text{out}} = 8\left[\left(\frac{1}{1 + N_C \cdot \frac{C_U}{C_0}} - \frac{N_R}{8}\right)T_{\text{CKV}} + \frac{3}{8}T_{\text{CKV}} - \Delta t_S\right] \tag{10}$$

where N_C is the CT code during the first discharge and N_R is the RT code during the third discharge. To explain the correlation between this output and the functional requirement in (1), the TAU execution cycle is divided into a reset state and three functional states—pre-discharge, snapshot, and time amplification (TA). Each of them realizes one term or coefficient in (1).

The execution cycle starts with the reset state, in which the SWC closes the relevant switches in the WTRs to charge all the capacitors (CT = max) to V_{init}. Then, the non-critical FREF (i.e., rising) edge disconnects the SWC switches and triggers the pre-discharge state, in which the TAU calculates and stores the Δt_S prediction term, $(1 - \phi_{\text{R,frac}}) \cdot T_{\text{CKV}}$. The prediction is realized by the weighted sum of three T_{CKV}'s, which are generated by sampling the CKV period and reflected on the width differences of the active-low SWD pulse-pairs. During the first SWD pulse-pair, the capacitive tuning code N_C (on CT) is applied to finely scale T_{CKV}. During the third one, the resistive tuning code N_R (on RT) scales T_{CKV} coarsely. The difference between these two scaled inputs realizes the $(1 - \phi_{\text{R,frac}}) \cdot T_{\text{CKV}}$ term in (1) with

$$\phi_{\text{R,frac}} = \frac{N_R}{8} + \left(1 - \frac{1}{1 + N_C \cdot \frac{C_U}{C_0}}\right). \tag{11}$$

Here, N_R ranges from 0 to 7, yielding the resolution of 1/8 in $\phi_{\text{R,frac}}$ tuning. Consequently, the N_C term needs only to cover the tuning range of $0 \sim 1/8$. Within such a narrow range, the nonlinearity in the mapping between N_C and $\phi_{\text{R,frac}}$ is insignificant and simple to compensate for. One may notice that (1) does not reflect the influence of the second discharge. In fact, this discharge introduces an extra offset of $3/8 \cdot T_{\text{CKV}}$ for metastability mitigation to be discussed in Section III-B1.

After these three discharges, TAU enters the snapshot state, in which the WTRs directly subtract the sampled Δt_S from the pre-stored prediction. This realizes the $-\Delta t_S$ term in (1). As a result, only the desired residue (substantially reflecting the DCO PN in the phase-locked state) remains in the TAU. Finally, in the TA state, the TAU outputs this residue as the time offset between CMP$_{\text{P}}$ and CMP$_{\text{N}}$ (Δt_{out}). During this process, the residue is also time-amplified by

$$G_{\text{TA}} = \frac{\tau_A}{\tau_S} = 8. \tag{12}$$

This gain factor corresponds to G_{TA} in (1) and is realized by manipulating the ratio between τ_A and τ_S, more specifically, the RT code during the TA and snapshot states. After generating the outputs, the TAU returns to the reset state, awaiting the next cycle.

III. Circuit-Level Implementation of TAU

A. TAU Sub-System Overview

Fig. 6(a) illustrates the implemented TAU together with the auxiliary circuits that control its behavior in each state defined in Section II-D. The PFD is actually realized in a more complex tri-mode in order to effectively support the three distinct functional states—pre-discharge, snapshot, and TA. The TAU is alternatively controlled by the global and local finite state machines (FSMs). Fig. 6(b) shows the active FSM in each TAU state, indicated by $\overline{\text{RST}_{\text{all}}}$, PDIS$_{\text{done}}$, and $\overline{\text{TA}_{\text{en}}}$. In the pre-discharge state, the local FSM is active. It interacts with the tri-mode PFD (through $\overline{\text{START}}$ and READY) to generate the first three inputs for the WTRs (pulse-pairs on SWD$_{\text{P}}$ and SWD$_{\text{N}}$). Meanwhile, the local FSM adjusts the weight for each input (through RT, CT, and SIGN), whose $\phi_{\text{R,frac}}$-dependent weight codes, i.e., N_R and N_C, are calculated by the RC encoder according to (11). Once the TAU processes the first three inputs, the local FSM terminates the pre-discharge state and activates the global FSM through PDIS$_{\text{done}} = 1$, which controls the TAU in the remaining states.

In the snapshot state, the global FSM captures Δt_S and transfers it to the TAU via CKRG$_{\text{P}}$ and CKRG$_{\text{N}}$. To mitigate the issue of potential metastability in the Δt_S sampling (see Section III-B1), an *anti-alignment* delay (between FREF and FREF') is added. In the TA state, the global FSM controls the local FSM to apply proper RT for G_{TA} and prepares the TAU for final output, both by setting $\overline{\text{TA}_{\text{en}}} = 0$. While waiting for the TAU output, the global FSM also launches CKU, a master

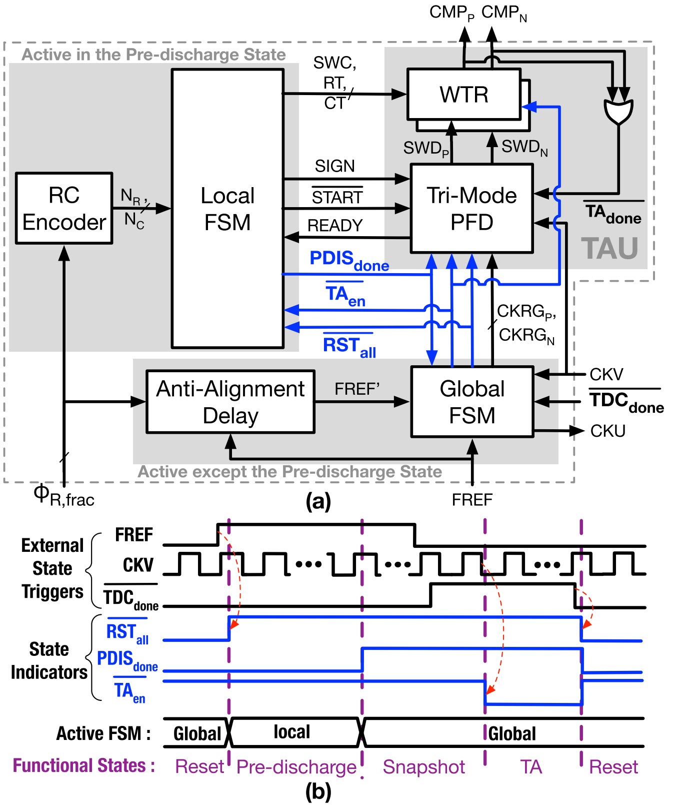


Fig. 6. (a) Simplified diagram of the TAU-centered sub-system (without calibration circuitry shown). (b) Timing diagram of the state transition (indicated by $\overline{\mathrm{RST}}_{\mathrm{all}}$, $\mathrm{PDIS}_{\mathrm{done}}$, and $\overline{\mathrm{TA}}_{\mathrm{en}}$).

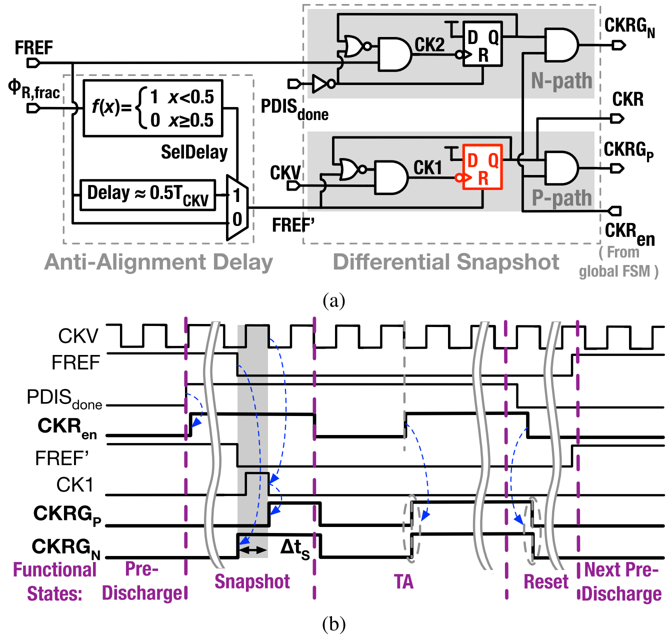


Fig. 7. Differential snapshot circuit: (a) schematic and (b) waveforms (for the case of $\phi_{\mathrm{R,frac}} \geq 0.5$).

clock of the overall PLL. After the TAU output is quantized by its subsequent TDC (indicated by $\overline{\mathrm{TDC}}_{\mathrm{done}}$ falling), the global FSM resets the overall TAU sub-system with $\overline{\mathrm{RST}}_{\mathrm{all}} = 0$. When this global reset is removed ($\overline{\mathrm{RST}}_{\mathrm{all}} = 1$, by the FREF rising), the local FSM will be activated again, starting the next execution cycle.

B. Implementation of the Global FSM

1) Differential Snapshot Circuit: In the snapshot state, the global FSM conveys the Δt_S information to the TAU via $\mathrm{CKRG_P}$ and $\mathrm{CKRG_N}$. Inside the global FSM, Δt_S is sampled by the differential snapshot circuit. As shown in Fig. 7, it contains two similar single-ended paths, modified from [14]. The P-path captures the first CKV falling edge after FREF'. To achieve this, FREF' first inactivates the reset on the main flip-flop (FREF' $= 0$) and releases CK1, the gated CKV. Once CKV falls, the main flip-flop asserts $\mathrm{CKRG_P}$. On the N-path, $\mathrm{CKRG_N}$ is asserted at the FREF falling edge (since $\mathrm{PDIS_{done}} = 1$ in the snapshot state). Ideally, the propagation delays on these two paths are canceled, so the time offset between the CKRGs equals that between FREF and CKV, which is Δt_S. One may also notice $\mathrm{CKR_{en}}$, the gating signal of CKRGs, in the differential snapshot circuit. It is scheduled by the global FSM (see Fig. 9) for two purposes: First, in the TA state, it launches the concurrent rising edges on the CKRGs to trigger the TAU output. Second, in the pre-discharge and reset states, it blocks activities on the CKRGs to avoid interfering with the tri-mode PFD.

The differential snapshot circuit can sample Δt_S accurately only if its N- and P-path propagation delays are properly canceled. However, in reality, the flip-flop metastability may corrupt this condition, thus distorting the sampled Δt_S. For example, in the P-path, the flip-flop's CK1-to-Q delay can dramatically increase when the reset removal (FREF' falling) is close to the subsequent critical clock edge (CK1 falling). This occurs with a certain probability (determined by the flip-flop's metastability window) in a fractional-N PLL mode because the time offset between the FREF and CKV edges (also, by extension, the offset between FREF' and CK1) distributes uniformly between 0 and T_{CKV}. In contrast, the N-path is free from this issue since its reset, the inverse of $\mathrm{PDIS_{done}}$, can be guaranteed to settle sufficiently earlier than CK2 (or FREF). Consequently, the P-path delay variation can reflect on the time offset between $\mathrm{CKRG_P}$ and $\mathrm{CKRG_N}$, thus adding uncertainty to the sampled Δt_S.

To avoid this flip-flop metastability issue, we add a conditional *anti-alignment* delay, either 0 or $T_{\mathrm{CKV}}/2$, between FREF' and FREF according to the Δt_S prediction [i.e., $(1 - \phi_{\mathrm{R,frac}}) \cdot T_{\mathrm{CKV}}$]. Consequently, the FREF' falling edge can be sufficiently separated from its neighboring CKV (strictly speaking, CK1) falling edge, and the flip-flop metastability will not occur. However, this variable delay will change the sampled Δt_S. For example, when FREF is close to its first subsequent CKV edge (i.e., small Δt_S prediction), FREF' is delayed by $T_{\mathrm{CKV}}/2$ to extend the separation. As a result, the second CKV edge after FREF, instead of the first one, is captured by the snapshot circuit, and T_{CKV} is added into the sampled Δt_S. In contrast, the sampled Δt_S is intact when its prediction is nominally large. This yields a non-monotonic mapping from $\phi_{\mathrm{R,frac}}$ to the sampled Δt_S, thus complicating the TAU control. To alleviate this, we add the $3/8T_{\mathrm{CKV}}$ offset during the 2nd discharge in the pre-discharge state (see Fig. 5). Since any type-II PLL always keeps a zero-mean input to the loop filter, this offset finally appears in the expected Δt_{S}

$$\Delta t_{\mathrm{S}} = \left(1 - \phi_{\mathrm{R,frac}}\right) \cdot T_{\mathrm{CKV}} + \frac{3}{8} T_{\mathrm{CKV}} \tag{13}$$

so the delay logic changes accordingly (see Fig. 7). This maintains a monotonic mapping between $\phi_{\mathrm{R,frac}}$ and the sampled Δt_{S} as (13), thus avoiding any complicated top-level time-offset controls.

To provide a better overview of this metastability mitigation mechanism, four boundary cases are examined in Fig. 8. From (a) to (d), these cases are arranged with increasing Δt_{S} (hence, decreasing $\phi_{\mathrm{R,frac}}$). In (a), FREF' is relatively close to the subsequent CKV. As Δt_{S} increases, FREF' separates from the subsequent CKV edge but gets closer to the precedent CKV edge until (b), right before the anti-alignment delay changes (controlled by SelDelay). At the moment SelDelay switches from 0 to 1 [see (c) when $\phi_{\mathrm{R,frac}} = 0.5$], FREF' is shifted by $T_{\mathrm{CKV}}/2$, thus closer to the subsequent CKV edge again, just as in (a). Then, as Δt_{S} increases, FREF' is gradually away from the subsequent CKV edge and closer to the precedent CKV edge until Δt_{S} reaches its maximum in (d), repeating the trend from (a) to (b).

There are two critical timing separations in these boundary cases. The first one is the minimum level of separation between FREF' and the subsequent CKV edge [see the light blue shaded area in (a) and (c)]. Increasing this separation helps to mitigate the linearity degradation due to metastability. The second is the minimum separation between FREF' and the precedent CKV edge [see the light red shaded area in (b) and (d)]. This separation is essential to avoid FREF' being caught up with the precedent CKV edge, which would cause the snapshot circuit to capture the wrong Δt_{S}. Thus, the value of this separation is not so critical as long as it does not cross zero.

Interestingly, the sum of these two critical separations equals $T_{\mathrm{CKV}}/2$ because the first one is set by the intentional offset for metastability mitigation (i.e., $3T_{\mathrm{CKV}}/8$ in our case), and the second one is set by $T_{\mathrm{CKV}}/2$ minus this intentional offset. It seems optimal to equally allocate $T_{\mathrm{CKV}}/2$ to these two separations, i.e., $T_{\mathrm{CKV}}/4$ for either. However, because the separation between FREF' and the subsequent CKV edge can cause the linearity issue, we prefer to assign more margin to it.

Although adding the offset of $3T_{\mathrm{CKV}}/8$ alleviates the metastability issue, it shifts the range of Δt_{S} from $(0, T_{\mathrm{CKV}}]$ to $(3T_{\mathrm{CKV}}/8, 11T_{\mathrm{CKV}}/8]$, thereby increasing the maximum Δt_{S} to $11T_{\mathrm{CKV}}/8$. To handle the increased Δt_{S}, the WTRs should adopt a larger R_0C_0 (see Section III-E), but this slows the discharge slew rate and degrades the noise performance (see Section V-C). This is a tradeoff between linearity (which may be degraded due to metastability) and noise. However, more advanced technology nodes will suffer less from this tradeoff because the flip-flops are faster with a narrower metastability window [32].

2) Time Amplification Control and Global Reset: Fig. 9 shows the overall global FSM, emphasizing the TA control logic and the global reset. The core of the TA control logic is a shift-register chain, whose outputs ($\mathrm{ST}\langle 2:0\rangle$) serve as a state variable, scheduling the TA-related actions: In the state of $\mathrm{ST}\langle 2:0\rangle = 3'\mathrm{b}001$, the global FSM notifies the local FSM to adjust RT for G_{TA}, alters the tri-mode PFD to the TA mode, and prepares the WTR comparator for the final output. All these actions are performed by pulling down $\overline{\mathrm{TA}_{\mathrm{en}}}$. When $\mathrm{ST}\langle 2:0\rangle = 3'\mathrm{b}011$, the tri-mode PFD is triggered for the final output by the rising $\mathrm{CKR}_{\mathrm{en}}$, which launches $\mathrm{CKRG}_{\mathrm{P}} = 1$ and $\mathrm{CKRG}_{\mathrm{N}} = 1$ in the differential snapshot circuit. The shift-register chain is clocked by a gated CKV, i.e., CKTA. It is activated after sampling Δt_{S} (indicated by CKR rising) and deactivated after triggering TAU output ($\mathrm{ST}\langle 2:0\rangle = 3'\mathrm{b}111$). The TA logic also launches the master clock for the PLL digital part (CKU) after triggering the TAU output. This helps protect the critical events (e.g., sampling Δt_{S} and launching the final output of TAU) from potential interferences due to the digital activity.

Once the output of TAU has been quantized (indicated by $\overline{\mathrm{TDC}_{\mathrm{done}}} = 0$ from the TDC), the global FSM asserts a global reset ($\overline{\mathrm{RST}_{\mathrm{all}}} = 0$). As a result, the TAU enters the reset state, waiting for the next TAU execution cycle (triggered by FREF rising).

C. Implementation of the Tri-Mode PFD

Fig. 10(a) shows details of the tri-mode PFD, whose three modes pair up with the three functional states of TAU. These modes are switched according to the TAU state indicators—$\overline{\mathrm{RST}_{\mathrm{all}}}$, $\mathrm{PDIS}_{\mathrm{done}}$, and $\overline{\mathrm{TA}_{\mathrm{en}}}$ [see Fig. 6(b)].

PFD Mode 1 is active in the pre-discharge state. The PFD core is driven then by the dedicated clock gating block, which releases the gated CKV clocks on $\mathrm{CKVG}_{\mathrm{P}}$ and $\mathrm{CKVG}_{\mathrm{N}}$ with one CKV cycle delay (when READY = 0). Once the CKVGs are released, the PFD core launches an active-low pulse-pair on $\mathrm{SWD}_{\mathrm{P}}$ and $\mathrm{SWD}_{\mathrm{N}}$, whose width difference is T_{CKV}. Fig. 10(b) illustrates a single SWD pulse-pair generation cycle. Once a cycle is triggered ($\overline{\mathrm{START}}$ falling, event marker ⟨1⟩), the flip-flop Q2 removes the reset on the output flip-flops Q1 and Q3 (RST = 0, ⟨2⟩), unsets the PFD idle flag (READY = 0, ⟨3⟩), and enables the CKV gating block to release the CKVGs successively (⟨4.1⟩ and ⟨4.2⟩). At the CKVGs' rising edges, the corresponding SWDs fall (⟨5.1⟩ and ⟨5.2⟩). Once both the SWDs become LOW, they are reset (⟨6⟩) to HIGH simultaneously (⟨7⟩). Consequently, the PFD outputs an active-low pulse-pair on the SWDs. Meanwhile, the SWD reset (⟨6⟩) also raises the PFD idle indicator (READY = 1, ⟨7*⟩), which is the check signal for the local FSM (see Fig. 12) to determine whether to start the next pulse-pair generation cycle (through $\overline{\mathrm{START}} = 0$, ⟨8⟩). In addition, as mentioned in Section II-B, the TAU needs to swap the leading-falling edges in the generated SWD pulse-pair when a negative weight is required. The SIGN signal (from the local FSM) controls this polarity by determining the earlier released CKVG. A question may arise whether the output flip-flops Q1 and Q3 can be disturbed by the activities on $\mathrm{CKRG}_{\mathrm{P}}$ and $\mathrm{CKRG}_{\mathrm{N}}$ in PFD Mode 1. According to Fig. 7, this cannot happen since the CKRGs are blocked by $\mathrm{CKR}_{\mathrm{en}} = 0$ in the pre-discharge state.

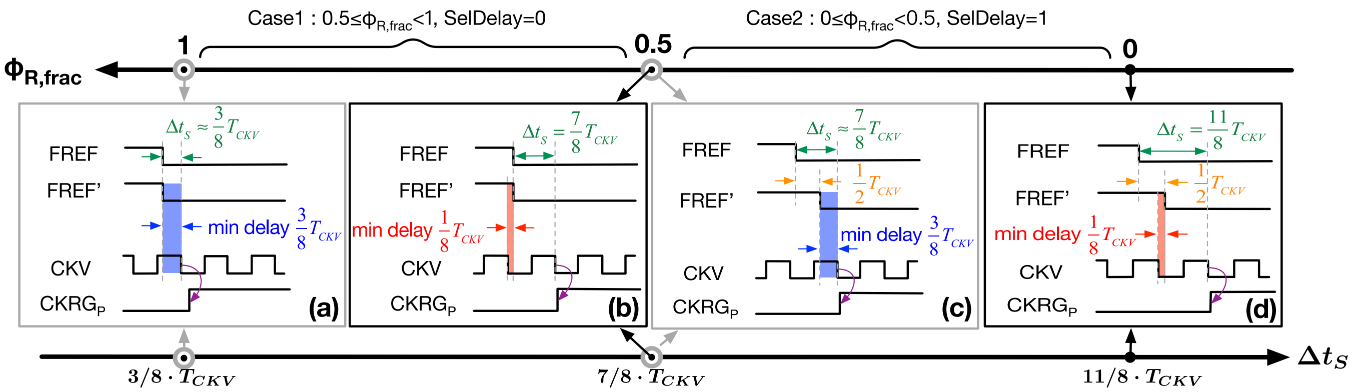


Fig. 8. Boundary cases of the metastability mitigation mechanism that prevents the insufficient separation between FREF' and the subsequent CKV edge [corresponding to CK1 in Fig. 7(a)].

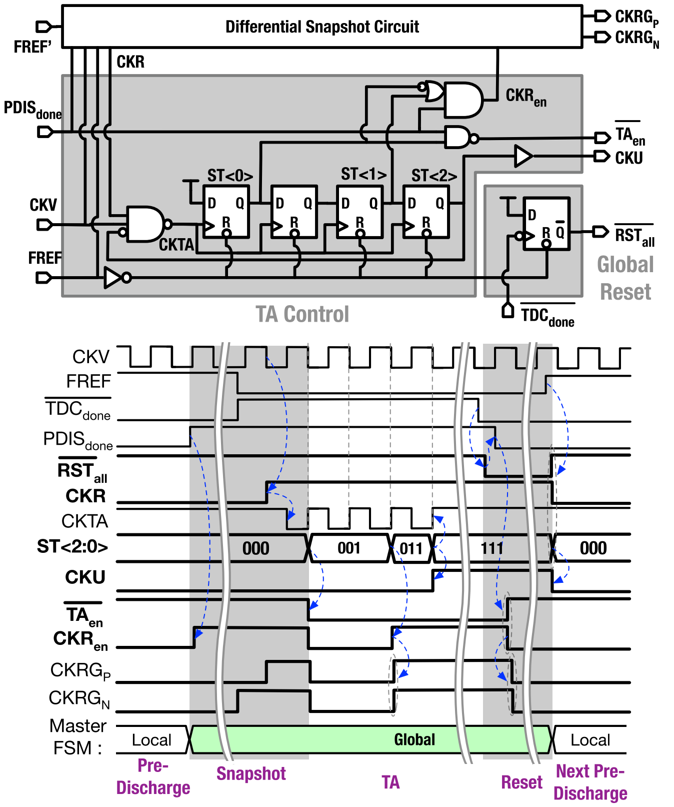
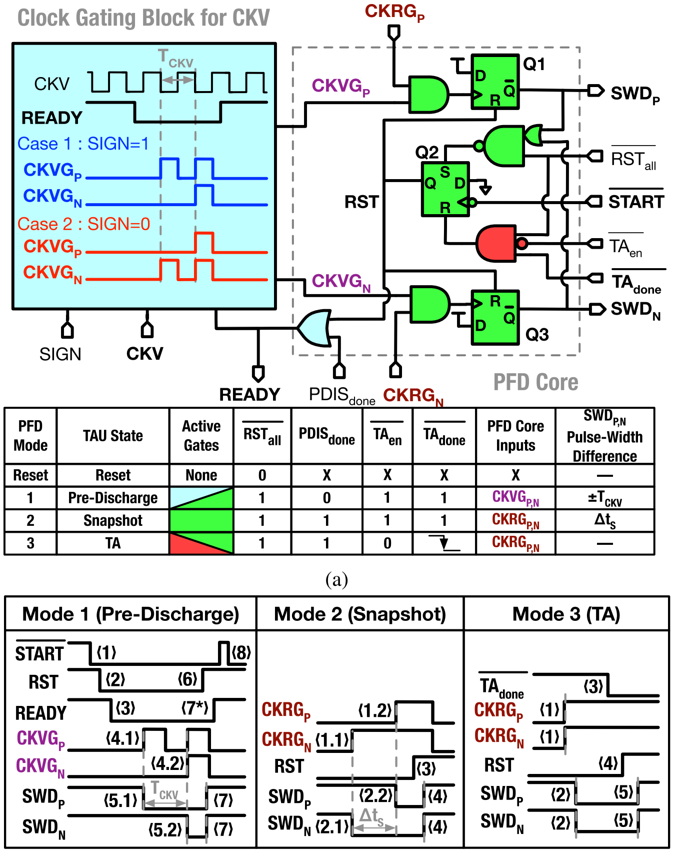


Fig. 9. Schematic and waveform diagrams of the global FSM.



PFD Mode	TAU State	Active Gates	$\overline{\mathrm{RST}}_{\mathrm{all}}$	$\mathrm{PDIS}_{\mathrm{done}}$	$\overline{\mathrm{TA}}_{\mathrm{en}}$	$\overline{\mathrm{TA}}_{\mathrm{done}}$	PFD Core Inputs	$\mathrm{SWD}_{\mathrm{P,N}}$ Pulse-Width Difference
Reset	Reset	None	0	X	X	X	X	—
1	Pre-Discharge	🟦🟩	1	0	1	1	$\mathrm{CKVG}_{\mathrm{P,N}}$	$\pm T_{\mathrm{CKV}}$
2	Snapshot	🟩	1	1	1	1	$\mathrm{CKRG}_{\mathrm{P,N}}$	Δt_S
3	TA	🟥🟩	1	1	0	↓	$\mathrm{CKRG}_{\mathrm{P,N}}$	—

(a)

(b)

Fig. 10. Tri-mode PFD: (a) simplified diagram and (b) its waveforms. In the reset flip-flop Q2, the R(eset) has higher priority than the S(et).

After the pre-discharge, the CKVGs are frozen at LOW by $\mathrm{PDIS}_{\mathrm{done}} = 1$. Then, the tri-mode PFD is driven by the CKRGs and behaves the same as the dual-mode PFD in the conceptual TAU of Fig. 3(d). Detailed waveforms are illustrated in Fig. 10(b): In PFD Mode 2 (paired with the snapshot state), the PFD converts the time difference between the CKRGs to the width difference of the SWD pulse-pair. In PFD Mode 3 (corresponding to the TA state), reset of the output flip-flops Q1 and Q3, i.e., RST, is initially disabled [by $\overline{\mathrm{TA}}_{\mathrm{en}} = 0$, note that $\overline{\mathrm{RST}}_{\mathrm{all}} = 1$ and $\overline{\mathrm{TA}}_{\mathrm{done}} = 1$ at this moment, and R(eset) has a higher priority than S(et) in flip-flop Q2]. Consequently, SWDs can remain at LOW (⟨2⟩) after being triggered by the CKRGs (⟨1⟩). The LOW-level SWDs keep discharging the WTRs. As soon as both WTRs output, a feedback signal [⟨3⟩, $\overline{\mathrm{TA}}_{\mathrm{done}} = \mathrm{CMP_P}+\mathrm{CMP_N} = 0$, as shown in Fig. 6(a) (upper right)] enables the reset (RST = 1, ⟨4⟩) so that the SWDs can recover HIGH level (⟨5⟩) in order to stop discharging the WTRs.

D. Implementation of the Local FSM

In the pre-discharge state, the local FSM controls the tri-mode PFD to generate the first three SWD pulse-pairs and applies proper weights to the WTRs. Each pulse-pair is generated through the interaction between the local FSM

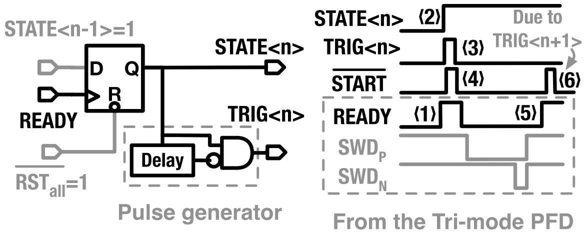
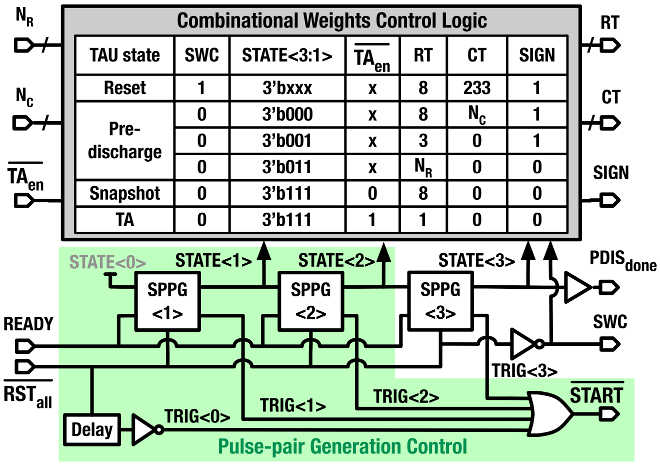


Fig. 11. SPPG logic.



Combinational Weights Control Logic

TAU state	SWC	STATE<3:1>	$\overline{\text{TA}}_{\text{en}}$	RT	CT	SIGN
Reset	1	3'bxxx	x	8	233	1
Pre-discharge	0	3'b000	x	8	N_C	1
	0	3'b001	x	3	0	1
	0	3'b011	x	N_R	0	0
Snapshot	0	3'b111	0	8	0	0
TA	0	3'b111	1	1	0	0

Fig. 12. Simplified diagram of the local FSM.

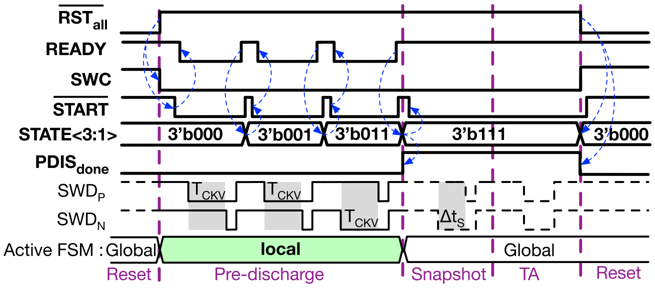


Fig. 13. Waveforms of the local FSM.

and the tri-mode PFD in a self-timed style, emulating the asynchronous SAR ADC [33]. Fig. 11 shows the detailed single pulse-pair generation (SPPG) logic. Two prerequisites are needed to activate the SPPG logic: the global reset released ($\overline{\text{RST}}_{\text{all}} = 1$) and the precedent (if existing) SPPG logic completed ($\text{STATE}\langle n-1\rangle = 1$). Once the tri-mode PFD becomes idle (READY = 1, ⟨1⟩), the SPPG cycle starts by raising its state indicator (STATE⟨n⟩ = 1, ⟨2⟩). Then, a trigger pulse is generated (on TRIG⟨n⟩, ⟨3⟩) to notify the tri-mode PFD to launch an SWD pulse-pair (through $\overline{\text{START}}$, ⟨4⟩, which sums the TRIG⟨n⟩'s from all the SPPG units in Fig. 12). Once the pulse-pair gets generated, the tri-mode PFD sets the idle flag again (READY = 1, ⟨5⟩), possibly starting the next SPPG cycle (⟨6⟩).

Fig. 12 sketches the overall local FSM, which cascades three SPPG units and sums their trigger pulses ($\overline{\text{START}} = \sum_{i=0}^{3} \text{TRIG}\langle i\rangle$) to launch the SWD pulse-pairs consequentially. The corresponding timing diagram in a complete TAU execution cycle is shown in Fig. 13. After activated by the global reset removal ($\overline{\text{RST}}_{\text{all}} = 1$), the local FSM disconnects the TAU's charging switch (SWC = 0) and triggers the tri-mode PFD (through the first $\overline{\text{START}}$ falling edge) to generate the first SWD pulse-pair. After that, the SPPGs interact with tri-mode PFD (through $\overline{\text{START}}$ and READY) to launch the remaining two SWD pulse-pairs (as shown in Fig. 11). Once "done" (indicated by the 3rd READY rising), the state of the TAU transitions from the pre-discharge to snapshot ($\text{PDIS}_{\text{done}} = 1$). Accordingly, the tri-mode PFD changes its mode. Then, at the local FSM's further request for pulse-pair generation (the 4th $\overline{\text{START}}$ falling), the tri-mode PFD merely removes its output reset, i.e., RST falls in Fig. 10(a), readying itself for processing Δt_S in the snapshot state.

The weight for each WTR discharge is controlled by the corresponding combinational logic in the local FSM (see Fig. 12), which translates the outputs of RC encoder (N_C and N_R) to the weight-control sequences (on RT, CT, and SIGN) according to the SWD pulse-pair indexes (STATE⟨3 : 1⟩) and certain TAU state indicators ($\overline{\text{TA}}_{\text{en}}$, and the inverted $\overline{\text{RST}}_{\text{all}}$, i.e., SWC). Note that the delay lines in the local FSM and SPPGs are realized with replica logic gates and routing of the corresponding weight control paths, in order to emulate the associated propagation delay. Therefore, these delays guarantee the corresponding discharges to be triggered (by $\overline{\text{START}}$ falling) after the weight control signals get settled down.

E. Implementation of the WTR

Fig. 4 shows the implemented WTR. The switching SR and SC units adopt dummy switches, roughly compensating their main switches' charge injection and clock feed-through in order to minimize the TAU's arithmetic accuracy degradation. Finer compensation is performed by a piecewise pre-distortion in the RC encoder (see Section VI-C). Considering that the overall TAU targets 10-bit accuracy, the WTR uses eight SR units and 223 SC units to realize the upper 3 bits and lower 7 bits, respectively. The over-designed 223 SC units provide enough redundancy for pre-distortion (or calibration).

A question might arise as to why the bottom plates of the SC units here are connected to power (V_{DD}) instead of ground. This is to avoid a situation where the bottom plate voltages of those disconnected SC units fall below ground after the discharge. This could occur if the bottom plates were initially connected to the ground and would result in reverse polarization of their switches, causing charge leakage, thus degrading the TAU's arithmetic accuracy.

The slicing comparator is modified from the threshold-crossing detector (TCD) in [34]. As shown in Fig. 14, the implemented slicer mainly consists of a gated inverter (PM2 and NM1) and a dynamic inverter (PM3, NM3, and NM4). The slicer is enabled (by $\overline{\text{TA}}_{\text{en}} = 0$) right before the final discharge of the WTR to avoid unnecessary power consumption due to the possible crowbar current (since V_C can be close to the threshold of the first-stage inverter before the final discharge). Once the slicer output is asserted

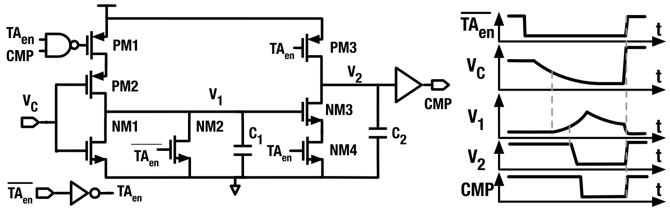


Fig. 14. Level-crossing slicer in the WTR: schematic and waveforms.

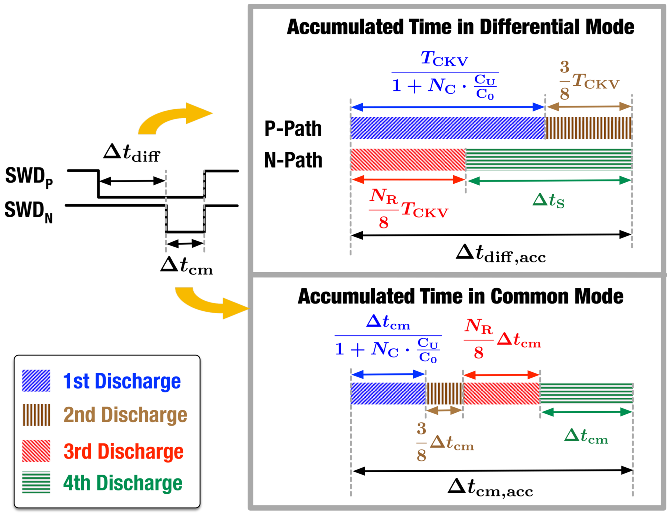


Fig. 15. Visualization of the equivalent discharge time that accumulates on the DWTRs during the four discharge-pulse-pairs in Fig. 5. Amounts of the discharge time refer to their equivalents in the snapshot state.

(CMP $= 0$), the first-stage inverter is gated off immediately to save power. Capacitors C_1 and C_2 help to suppress the output jitter [34]. The cross detection threshold of this slicer, V_{th}, is dominated by that of the first-stage inverter, which drifts with PVT variations. Fortunately, the differential arrangement helps to cancel the influence of V_{th} drift common to both paths. V_{th} mismatch between the differential paths mainly causes a constant output offset, which is automatically compensated by the loop dynamic of a type-II PLL.

Considering the constraint in Section II-B stating that V_C should be higher than V_{th} after the (W)TR processes all the time inputs, one may wonder how to properly choose V_{init}, V_{th}, and the R & C values of the WTR to satisfy this constraint. From the circuit perspective, these four physical parameters determine the upper limit of the discharge duration that a WTR *can* handle, i.e., Δt_{lim}. From the system perspective, the time processing details in Fig. 5 determine the maximum discharge duration the TAU *should* handle, i.e., Δt_{max}. As long as $\Delta t_{max} < \Delta t_{lim}$, V_C would never fall below V_{th} after all the inputs get processed. In this way, the four physical parameters of the WTR are constrained. Next, we calculate Δt_{lim} and Δt_{max} separately. Note that, in the analysis below, all the discharge durations are referred to as their corresponding equivalents in the snapshot state, i.e., resulting in the same amount of V_C drop if discharging C_0 through $R_0/8$. This is because the primary goal of the TAU is to cancel Δt_S, which is processed in the snapshot state. Δt_{lim} can be determined by discharging C_0 from V_{init} to V_{th} through $R_0/8$

$$\Delta t_{lim} = \frac{R_0 C_0}{8}\ln\left(\frac{V_{init}}{V_{th}}\right). \tag{14}$$

To analyze Δt_{max}, Fig. 15 depicts the equivalent discharge time of the DWTRs. Each SWD pulse-pair contains a differential component Δt_{diff} and a common-mode component Δt_{cm}. The former is the explicit time input to be processed, i.e., T_{CKV} or Δt_S, depending on the state of the TAU; the latter results from the PFD reset delay. The influences of these two components should be considered separately.

Considering that the time signals on the P and N paths will cancel out, the maximum accumulated duration in the differential mode can be estimated by inspecting the P-path as

$$\max\left(\Delta t_{diff,acc}\right) = \left[\max\left(\frac{1}{1+N_C\cdot C_U/C_0}\right)+\frac{3}{8}\right]\cdot T_{CKV} = \frac{11}{8}\cdot T_{CKV} \tag{15}$$

which is obtained at $N_C = 0$. For the common-mode discharge, the max accumulated duration is

$$\max\left(\Delta t_{acc,cm}\right) = \left[\max\left(\frac{1}{1+N_C\cdot C_U/C_0}\right)+\frac{3}{8} + \max\left(\frac{N_R}{8}\right)+1\right]\cdot\Delta t_{cm} = \frac{26}{8}\cdot\Delta t_{cm} \tag{16}$$

which is achieved at $N_C = 0$ and $N_R = 7$. Summing $\max(\Delta t_{acc,diff})$ and $\max(\Delta t_{acc,cm})$ yields Δt_{max}. By substituting (14)–(16) into $\Delta t_{max} < \Delta t_{lim}$, the minimum required product of $R_0 \times C_0$ can be constrained as

$$R_0 C_0 > \frac{11T_{CKV}+26\Delta t_{cm}}{\ln(V_{init}/V_{th})}. \tag{17}$$

F. Implementation of the RC Encoder

The *RC* encoder assists the local FSM with the weight control by mapping $\phi_{R,frac}$ to N_C and N_R, which are, respectively, the CT code at the first discharge and the RT code at the third discharge (see Fig. 5). According to (11), the mapping from $\phi_{R,frac}$ to N_R is linear. Considering that N_R is responsible for the coarse tuning, it is simply obtained by truncation

$$N_R = \lfloor 8\cdot\phi_{R,frac}\rfloor. \tag{18}$$

Then, N_C handles the residue phase

$$\phi_{CT} = \phi_{R,frac} - \frac{N_R}{8} = 1 - \frac{1}{1+N_C\cdot\frac{C_U}{C_0}}. \tag{19}$$

Accurate mapping from ϕ_{CT} to N_C is nonlinear and rather complex, but it can be approximated with Taylor series considering that ϕ_{CT} is merely a small residue (<1/8) after the coarse tuning

$$N_C = \frac{C_0}{C_U}\cdot\left(\frac{1}{1-\phi_{CT}}-1\right) = \frac{C_0}{C_U}\cdot\left[\phi_{CT}+\phi_{CT}^2+o(\phi_{CT})\right] \tag{20}$$

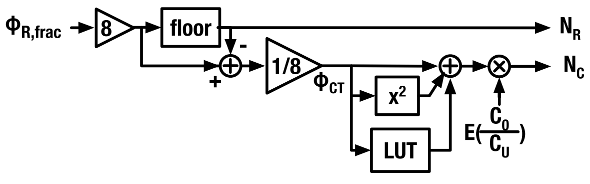


Fig. 16. Implementation diagram of the *RC* encoder.

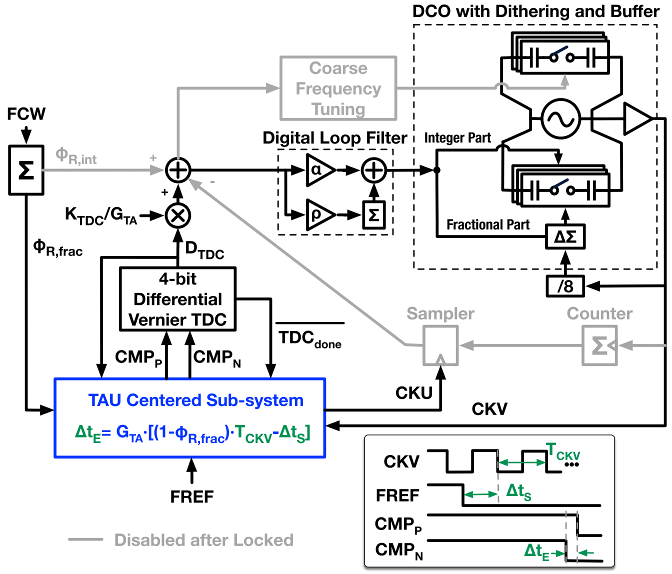


Fig. 17. Top-level diagram of the proposed PLL.

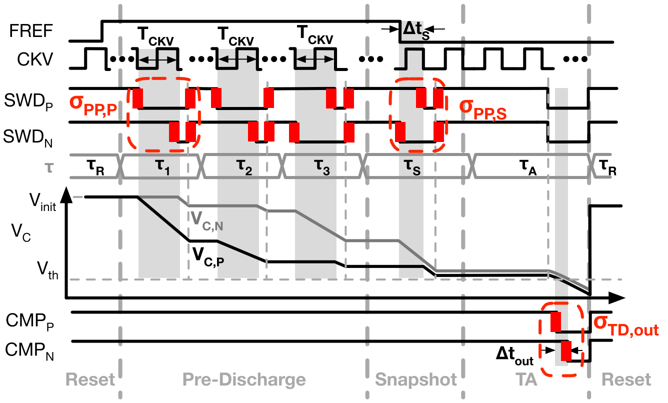


Fig. 18. Time-domain noise injected into the DWTRs.

where the dominant nonlinearity is handled by ϕ_{CT}^2, and higher order errors are compensated by $o(\phi_{\text{CT}})$. Fig. 16 illustrates the implemented *RC* encoder. The path from $\phi_{\text{R,frac}}$ to N_{R} reflects (18). Equation (20) is realized by the path from ϕ_{CT} to N_{C}, where a sparse lookup table (LUT) stores the high-order error $o(\phi_{\text{CT}})$, and $\mathbf{E}(C_0/C_{\text{U}})$ estimates the fabricated capacitance ratio C_0/C_{U}.

IV. Implemented PLL

The proposed TAU sub-system is incorporated into the fractional-N PLL shown in Fig. 17. The TAU extracts the time error Δt_{E}, mainly due to the DCO PN, by canceling Δt_{S} with its prediction. Unlike the DTC-based or voltage-domain methods, which cancel Δt_{S} with fixed time resolution, the TAU has a fixed *phase* resolution of $2\pi/1024$ as it scales the carrier period T_{CKV} with the 10-bit accuracy. The output of the TAU is quantized by a 4-bit differential TDC, whose overall architecture is quite similar to that in [11]. However, the sub-TDC for each differential path was replaced by a vernier counterpart in [34] in order to achieve a fine resolution of 1.9 ps. Considering the TAU's time amplification gain $G_{\text{TA}} = 8$, the equivalent TDC quantization resolution is finer than 240 fs, thus negligible for the PLL in-band PN. In parallel with the TAU-based phase error tracking path, there is also a counter path assisting the frequency (re)locking, which can be turned off to save power once the PLL is locked. Similar to in [6], [20], and [35], the counter path could be "instantaneously" woken up when the PLL gets unlocked as detected by a range detector in TDC. The DCO is implemented using an *LC* tank and a complementary cross-coupled pair as in [36]. It covers the oscillation frequency range from 2.6 to 4.1 GHz. The frequency tuning is achieved by SC banks, with the finest frequency resolution varying from 70 to 290 kHz, depending on the oscillation frequency. To reduce its PN contribution, the frequency resolution is dithered by a $\Delta\Sigma$ modulator (DSM), operating at $1/8$ DCO's resonant frequency.

V. Noise/Jitter Analysis

As the TAU adopts the DWTRs to perform time-domain signal processing, all the noise generated within the TAU sub-system will be eventually reflected in the differential output as timing variance. The noise sources are categorized into two types: the time-domain noise, which constitutes the SWD jitter and is added to WTRs in conjunction with the time-domain inputs, and voltage noise, which originates inside the WTRs. Each noise type shows a distinctive transfer function at the TAU output.

A. Time-Domain Noise

Fig. 18 depicts the time-domain noise presenting as jitter on the SWD edges. During the pre-discharge and snapshot states, the jitter that belongs to the same SWD pulse-pair is clustered as a pulsewidth difference variance, σ_{PP}. σ_{PP}'s in the pre-discharge and snapshot states are further distinguished as $\sigma_{\text{PP,P}}$ and $\sigma_{\text{PP,S}}$, respectively. σ_{PP}'s are injected into the DWTRs "riding" on top of their time-domain inputs to finally appear at the TAU output along the corresponding outputs. Therefore, the TAU's signal processing function of (10) also applies to σ_{PP}. Moreover, consider the two facts: $\sigma_{\text{PP,P}}$ and $\sigma_{\text{PP,S}}$, are added to T_{CKV} and Δt_{S}, respectively; the factor of 8 in (10) results from the time-amplification gain $G_{\text{TA}} = 8$ [see (12)]. Consequently, we obtain the code-dependent TAU output variance resulting from the time-domain noise

$$\sigma_{\text{TD,out}}^2(N_{\text{C}}, N_{\text{R}}) = {G_{\text{TA}}}^2 \cdot \left\{ \left[\left(\frac{1}{1 + N_{\text{C}} \cdot C_{\text{U}}/C_0} \right)^2 + \left(\frac{N_{\text{R}}}{8} \right)^2 + \left(\frac{3}{8} \right)^2 \right] \cdot \sigma_{\text{PP,P}}^2 + \sigma_{\text{PP,S}}^2 \right\}. \tag{21}$$

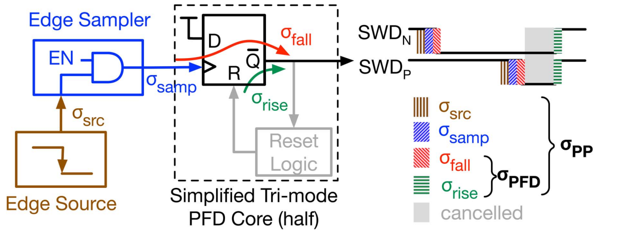


Fig. 19. Jitter contributors of an SWD pulse-pair. Note that only half of the PFD core is shown here, so σ_{PFD} consists of σ_{fall} and σ_{rise} contributions on both paths, yielding $\sigma_{\text{PFD}}^2 = 2\sigma_{\text{fall}}^2 + 2\sigma_{\text{rise}}^2$.

The N_{C}- and N_{R}-related coefficients represent $\phi_{\text{R,frac}}$ [see (11)], which uniformly distributes between 0 and 1 in fractional-N channels; thus, they can be averaged accordingly. This yields the average TAU output variance

$$\overline{\sigma_{\text{TD,out}}^2} \approx G_{\text{TA}}{}^2 \cdot \left(1.3\sigma_{\text{PP,P}}^2 + \sigma_{\text{PP,S}}^2\right). \tag{22}$$

B. Circuit-Level Contributors of Time-Domain Noise

Up until now, σ_{PP} has been treated as top-level composite noise. In this section, we break it down into circuit-level contributors so that we can estimate $\overline{\sigma_{\text{TD,out}}^2}$ by combining the simulated jitter of each sub-circuit. According to Fig. 19, three physical mechanisms contribute to σ_{PP}. The first is the original edge source that triggers the SWD falling edges, i.e., CKV or FREF. Its edges determine the SWD pulsewidth difference. Correspondingly, the edge source adds its jitter σ_{src} to σ_{PP}. The second σ_{PP} contributor is a conceptual edge-sampler, which samples the time information from the edge source and transfers it to the tri-mode PFD core. For example, in the snapshot state, it represents the differential snapshot circuit (see Fig. 7), which samples Δt_{S} from CKV and FREF. To realize the required functions, the edge samplers usually block the unwanted edges and pass the desired ones. Thus, the edge sampler smears out the desired edges during the propagation. Consequently, the edge sampler adds its jitter σ_{samp} to the SWD falling edges. The last σ_{PP} component is σ_{PFD}, i.e., width difference variance of the SWD pulse-pair due to the tri-mode PFD core, which launches the pulse-pair and contributes noise to both the SWD falling and rising edges. Since the PFD reset logic is common for the differential paths, its noise contribution is canceled in the final pulsewidth difference [37]. Therefore, only the output flip-flops degrade σ_{PFD}. Finally, σ_{PP} is broken down to

$$\sigma_{\text{PP}}^2 = 2\sigma_{\text{src}}^2 + 2\sigma_{\text{samp}}^2 + \sigma_{\text{PFD}}^2 \tag{23}$$

where factor 2 indicates that the edge jitter adds to both SWD paths.

For $\sigma_{\text{PP,P}}$, i.e., σ_{PP} in the pre-discharge state, its edge source is the CKV clock with jitter of σ_{CKV}, and the edge sampler is the CKV gating block in the tri-mode PFD with jitter of σ_{CKVG}. Therefore, $\sigma_{\text{PP,P}}$ is detailed as

$$\sigma_{\text{PP,P}}^2 = 2\sigma_{\text{CKV}}^2 + 2\sigma_{\text{CKVG}}^2 + \sigma_{\text{PFD}}^2. \tag{24}$$

For $\sigma_{\text{PP,S}}$, i.e., σ_{PP} in the snapshot state, its edge source contains both the CKV and FREF clocks, and the edge sampler is the differential snapshot circuit with jitter of σ_{snap} on either path. Therefore, the $\sigma_{\text{PP,S}}$ breakdown is

$$\sigma_{\text{PP,S}}^2 = \sigma_{\text{CKV}}^2 + 2\sigma_{\text{snap}}^2 + \sigma_{\text{PFD}}^2. \tag{25}$$

The coefficient of σ_{CKV}^2 is 1 since the CKV clock only launches one SWD falling edge. Although FREF triggers the other SWD falling edge, its jitter is expediently ignored here since it is usually considered as reference noise in the PLL systems. Substituting (24) and (25) into (22), we have

$$\overline{\sigma_{\text{TD,out}}^2} \approx G_{\text{TA}}{}^2 \cdot \left(3.6\sigma_{\text{CKV}}^2 + 2.6\sigma_{\text{CKVG}}^2 + 2\sigma_{\text{snap}}^2 + 2.3\sigma_{\text{PFD}}^2\right). \tag{26}$$

Note that σ_{CKV} includes only the jitter of the DCO's buffer since the intrinsic DCO PN has the nature of wander (i.e., accumulated jitter) [38], and so it must be accounted for separately.

C. Voltage Noise

In the TA state, the DWTRs convert their internal voltages into the time difference at the output. As such, any internal noise voltage will be manifested as time difference variance $\sigma_{\text{VD,out}}$. Two types of noise voltages dominate $\sigma_{\text{VD,out}}$—KT/C noise on the fixed capacitor C_0 and the noise voltage of the first-stage slicing comparator (see Fig. 14). For either WTR, its output jitter due to the KT/C noise is estimated as

$$\sigma_{\text{KT/C}}^2 = \frac{kT}{C_0} \cdot \frac{1}{k_{\text{th1}}^2} \tag{27}$$

where k is Boltzmann's constant, T is the absolute temperature, and k_{th1} is the slope of the C_0 discharge curve when it crosses V_{th1}, the threshold voltage of the first-stage cross comparator. With the windowed integral theory in [39], the first-stage cross comparator approximately degrades the WTR output jitter by

$$\sigma_{\text{cmp}}^2 = \frac{\sqrt{2}kT\gamma}{\sqrt{V_{\text{th2}} \cdot k_{\text{th1}}^3 \cdot g_{\text{m,eq}} \cdot C_1}} \tag{28}$$

where $g_{\text{m,eq}}$ is the equivalent transconductance combination of PM2 and NM1, C_1 is the load capacitance of PM2 and NM1, γ is the excess noise factor, and V_{th2} is the threshold voltage of the second stage of the cross comparator. Consequently, the TAU's output variance resulting from the voltage-domain noise is roughly

$$\sigma_{\text{VD,out}}^2 = 2 \cdot \left(\sigma_{\text{KT/C}}^2 + \sigma_{\text{cmp}}^2\right) \tag{29}$$

where factor 2 accounts for the differential operation.

D. TAU's Input-Referred Noise and Its Contribution to PLL's Phase Noise

Summing $\overline{\sigma_{\text{TD,out}}^2}$ and $\sigma_{\text{VD,out}}^2$ estimates $\sigma_{\text{TAU,out}}^2$, the overall time difference variance at the TAU output. Yet, we prefer to use the input-referred jitter for the PLL PN analysis, especially at the FREF side, e.g., [15] and [40]. According to (10) and (12), the transfer gain from FREF-related input, i.e., Δt_{S}, to TAU's output is $G_{\mathrm{TA}} = 8$. Therefore, $\sigma^2_{\mathrm{TAU,out}}$ is divided by ${G_{\mathrm{TA}}}^2$ to derive the TAU's input-referred jitter

$$\sigma^2_{\mathrm{TAU,in}} \approx 3.6\sigma^2_{\mathrm{CKV}} + 2.6\sigma^2_{\mathrm{CKVG}} + 2\sigma^2_{\mathrm{snap}} + 2.3\sigma^2_{\mathrm{PFD}} + \frac{\sigma^2_{\mathrm{KT/C}} + \sigma^2_{\mathrm{cmp}}}{32}. \quad (30)$$

Since the thermal noise dominates $\sigma^2_{\mathrm{TAU,in}}$, the noise spectrum can be assumed to uniformly spread over the reference frequency range f_{REF}. According to Staszewski and Balsara [15], this jitter power spectral density can be normalized to the PN spectrum by multiplying $(2\pi f_{\mathrm{CKV}})^2$, where f_{CKV} is the PLL output frequency. After getting attenuated by the closed-loop transfer function of the PLL, i.e., $H_{\mathrm{cl}}(f)$, $\sigma_{\mathrm{TAU,in}}$ contributes to the overall PLL PN by

$$\mathcal{L}_{\mathrm{TAU}}(f) = \frac{\sigma^2_{\mathrm{TAU,in}}}{f_{\mathrm{REF}}} \cdot (2\pi f_{\mathrm{CKV}})^2 \cdot |H_{\mathrm{cl}}(f)|^2. \quad (31)$$

VI. Nonlinearity Analysis

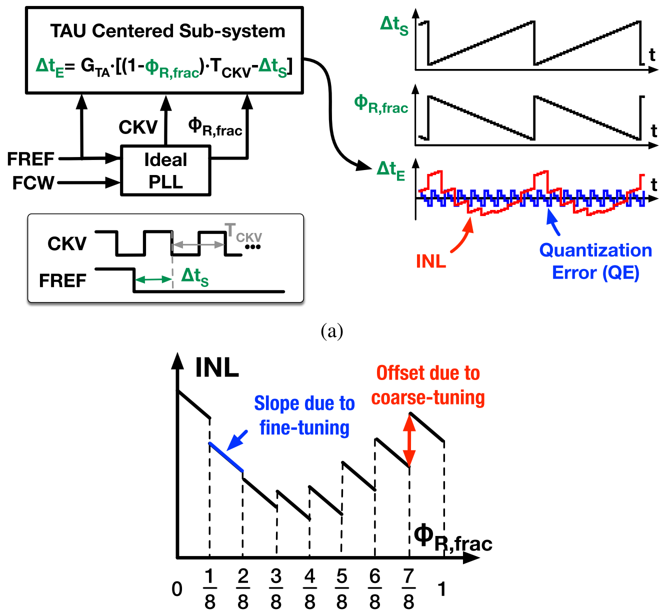
A. INL Characterization and Degradation Mechanism

Generally, a nonlinearity of a typical mixed-signal circuit (e.g., DAC and DTC) is characterized by an integral nonlinearity (INL) representing a deviation between the practical and ideal outputs across the input. However, this is inapplicable for TAU as it needs to handle multiple time-domain and digital inputs. However, if the scope is narrowed down to the time-offset cancellation case in a type-II PLL system, the TAU's INL can be well-defined. Consider the corresponding behavior of TAU described in (1). Δt_{S} is the time offset to be canceled, so it can be regarded as an ideal target, equivalent to the ideal output of a DTC. $(1 - \phi_{\mathrm{R,frac}}) \cdot T_{\mathrm{CKV}}$ is the generated term to cancel with Δt_{S} and, thus, can be treated as the counterpart of the actual DTC output. Therefore, the cancellation residue Δt_{E} reflects the TAU's nonidealities.

A conceptual testbench to measure the TAU's INL is illustrated in Fig. 20(a). Two phase-locked clocks, i.e., CKV and FREF, and the digital control target, i.e., $\phi_{\mathrm{R,frac}}$, are input to the TAU sub-system [similar as in Fig. 6(a)], emulating the inputs to the TAU in the proposed PLL. Under such an arrangement, the TAU can get a stable time base of T_{CKV}, a sequence of incremental Δt_{S} ramps, and the corresponding $\phi_{\mathrm{R,frac}}$, which scales the T_{CKV} to accurately cancel Δt_{S}. In the ideal case with no analog impairments, the cancellation residue Δt_{E} would reflect the TAU's quantization error (QE), which can be precisely estimated based on the *RC* encoder structure in Fig. 16. However, if the TAU's nonlinearity is included, Δt_{E} will further reflect the INL. Therefore, we can estimate the TAU's INL versus $\phi_{\mathrm{R,frac}}$ as

$$\mathrm{INL}(\phi_{\mathrm{R,frac}}) = \left[\frac{\Delta t_{\mathrm{E}}(\phi_{\mathrm{R,frac}})}{G_{\mathrm{TA}} \cdot T_{\mathrm{CKV}}} - \mathrm{QE}(\phi_{\mathrm{R,frac}})\right] \times 2^{10} \quad (32)$$

where $\mathrm{QE}(\phi_{\mathrm{R,frac}})$ is the quantization error in the same scale as $\phi_{\mathrm{R,frac}}$. After being divided by G_{TA}, Δt_{E} refers to the TAU's time input on the Δt_{S} side. This excludes the influence of time amplification, making the INL comparable with that of other time-offset cancellation circuits, such as DTCs. In addition, the multiplication by 2^{10} scales the unit of INL to the LSB of a 10-b converter, which is the case of the implemented TAU. (The unit before the scaling by 2^{10} is 1, i.e., characterizing the full range of T_{CKV} with $0 \sim 1$.)



(a)

(b)

Fig. 20. Characterization of the TAU's INL: (a) principle and (b) conceptually expected INL curves.

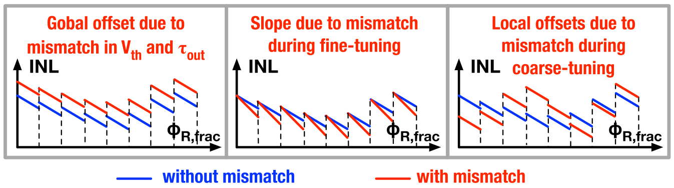


Fig. 21. INL curve of the TAU shaped by component mismatch.

Fig. 20(b) sketches a conceptually expected INL curve of the TAU. It exhibits a piecewise linear shape due to the TAU's coarse-fine tuning strategy. The eight segments coincide with the 3-b coarse resistive tuning. The vertical offset of each segment results from the nonideality of SR bank units, e.g., charge injection, clock feedthrough, and unit mismatch. The characteristic inside each segment is mainly correlated with the fine capacitive tuning. For example, the slope of each segment results from the C_0/C_{U} estimation error in (20) and the charge injection of SC-bank units. Since the fine-tuning is determined only by N_{C} during the first discharge (see Fig. 5), which is actually irrelevant for the subsequent coarse tuning behavior, the slopes of all the segments are almost identical.

One may wonder how the INL curve changes in face of a mismatch between the DWTRs. In fact, the overall piecewise linear feature would remain similar to that in Fig. 20(b), but the offsets and slope values of each segment would change. This can be analyzed by inspecting each term in (5) that describes the WTR function. First, consider the offset term $\tau_{\mathrm{out}} \cdot \ln(V_{\mathrm{init}}/V_{\mathrm{th}})$, which is supposed to be canceled out in

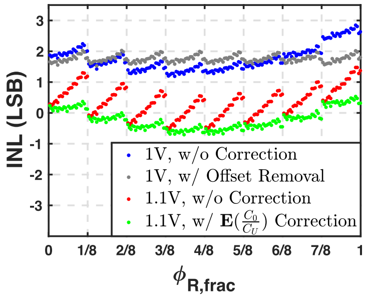


Fig. 22. Simulated INL of TAU at the supply of 1 and 1.1 V.

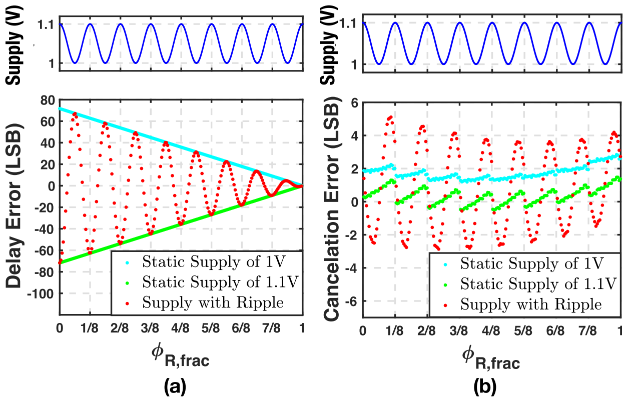


Fig. 23. (Equivalent) delay error under the sinusoidal supply fluctuating between 1 and 1.1 V. (a) Estimated from a virtual DTC emulating the resolution drift behavior in [41] and (b) simulation results of TAU.

the differential output. The mismatches in V_{th} and τ_{out}, i.e., the threshold voltage of the level-crossing comparator and the RC time constant during the final discharge, would result in a cancellation error, which globally offsets the overall INL curve [see Fig. 21 (left)]. As for each of the weighted terms, i.e., $\tau_{\mathrm{out}}/\tau_i \cdot \Delta t_i$, mismatches in the corresponding discharge RC time constants, i.e., τ_{out} and τ_i, would introduce error in the Δt_i scaling. Here, the mismatch of the SR unit dominates that of the RC time constants since the capacitive mismatch can be addressed by properly sizing the SC units [7]. The detailed effects due to this scaling error are case-dependent. For example, the scaling error would vary the slopes of all segments by the same amount if it occurred in the fine-tuning discharge (see Fig. 21 middle) because this discharge adopts a fixed SR configuration (RT = 8), and the corresponding mismatch introduces a fixed gain error to all the target scaling factors. In contrast, the scaling error would randomly offset each segment if it happened in the coarse-tuning discharge (see Fig. 21 right) since the error due to mismatch is N_{R}-dependent.

B. Simulated INL

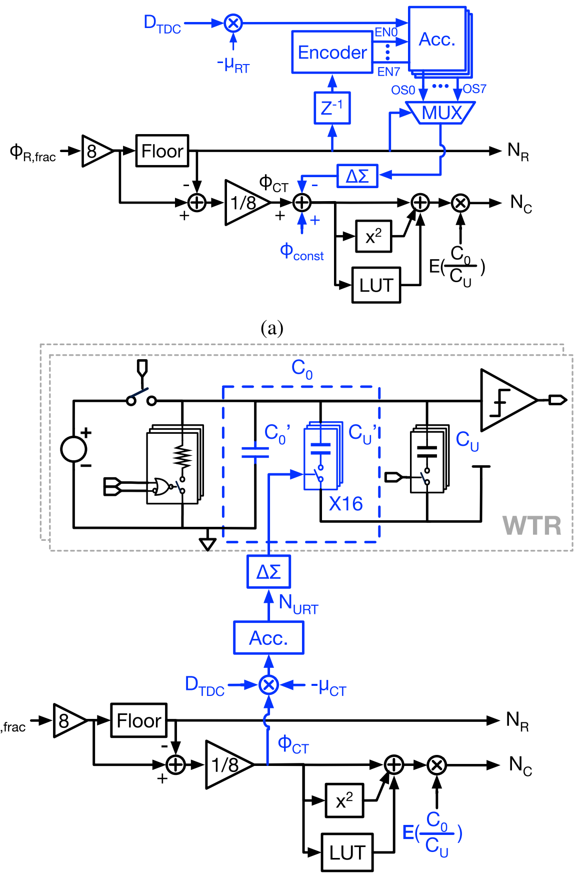
Fig. 22 shows the INL curve of TAU extracted from post-layout simulations. Under a 1-V supply (the nominal supply of transistors used in the implemented TAU), the INL is 1.7 LSB, corresponding to 0.17% of the full range. This is better than the DTC INL of 0.4% in [42] but worse than that of 0.09% in [41] (both from simulations). The TAU's INL is mainly degraded by the offsets between the coarse-tuning segments, reflecting the contribution from the charge injection of SR units. The INL could be improved to 0.5 LSB if the relative offsets were removed by calibration.

The INL under 1.1-V supply is also shown in Fig. 22. The slope of each segment increases significantly, thus degrading the INL to 2 LSB. The increased slope can be attributed to the nonlinear parasitic capacitance, which varies with supply, thus introducing more error to the estimated capacitance ratio in the RC encoder, i.e., $\mathbf{E}(C_0/C_{\mathrm{U}})$. After adjusting $\mathbf{E}(C_0/C_{\mathrm{U}})$, the slopes are essentially corrected, and so the INL drops to 1.2 LSB, which is 0.12% of the full range and the same as the DTC INL under 1.1 V in [41].

One may question the advantage of TAU given its apparent lack of superiority in the INL characteristics over those in the best-in-class DTCs, such as [41]. In fact, the INLs presented so far were simulated under ideal constant supply conditions and reflect only the "static" nonlinearity. In practice, the DTC delay is easily disturbed by instantaneous supply fluctuations and, thus, suffers from certain "dynamic" nonlinearity. For this reason, Wu *et al.* [7], Markulic *et al.* [22], and Santiccioli *et al.* [43] report significant efforts on stabilizing the supply.

This supply-related nonlinearity issue is examined with a 10-b virtual DTC example emulating the resolution drift behavior in [41]. The reported DTC resolution changes (becomes finer) by 14% when the supply increases from 1 to 1.1 V. Therefore, if the estimated DTC gain, K_{DTC}, is not adjusted accordingly, the DTC output delay would exhibit an error that is linearly proportional to the expected value. Fig. 23(a) shows the trend lines of the expected delay error of this reference DTC under the supply of 1 and 1.1 V, with the expected K_{DTC} (used for converting the expected delay to the DTC control word) frozen at the mean value of these two cases. The two trend lines are characterized under a test bench similar to Fig. 20(a), so they converge to 0 at $\phi_{\mathrm{R,frac}} = 1$, corresponding to the expected delay of 0, and reach the maximum amplitude at $\phi_{\mathrm{R,frac}} = 0$. One may doubt the efficacy of freezing the estimated K_{DTC} since a background calibration can constantly track the K_{DTC} drift. However, the calibration might be too slow to respond to fast supply disturbances. Fig. 23(a) shows a case with such a fast supply ripple, which sinusoidally fluctuates between 1 and 1.1 V, in synchronicity with $\phi_{\mathrm{R,frac}}$. The corresponding delay error of the virtual DTC will oscillate between the two aforementioned trend lines, and the peak-to-peak error can be up to 140 LSB.

For comparison, the Δt_{S} cancellation error of the TAU is simulated under the same supply ripple condition. According to Fig. 23(b), the peak-to-peak error is merely ~8 LSB. This benefits from the operating principle of scaling the "golden" time base and indicates the TAU would show stronger immunity to aggressors and better "dynamic" linearity compared with the DTC. One may wonder why the cancellation error of the TAU in face of the supply ripple exceeds the boundaries set by the INL curves under the stable supply cases (i.e., at 1 and 1.1 V). This comes from our specific WTR implementation,



(a)

(b)

Fig. 24. Foreground piecewise calibration for the INL of TAU: (a) Offset calibration for each coarse-tuning segment and (b) calibration for the fine-tuning slope.

where the bottom plates of the SC units are connected to V_{DD} (see Fig. 4). The supply ripple will affect the internal voltage of the WTRs (i.e., V_{C}) through the conducting SC units and parasitic switch capacitance, thus ultimately degrading the INL.

C. INL Calibration

According to Fig. 20(b), the INL of TAU is dominated by the coarse-tuning offsets and fine-tuning slope, correlated with N_{R} and ϕ_{CT} in Fig. 16, respectively. To combat the INL degradation relevant to these two sources, a piecewise calibration emulating [44] is added to supplement the *RC* encoder. The calibration operates when the PLL is locked by observing the TDC output, i.e., D_{TDC}. As shown in Fig. 24, the calibration consists of two parallel paths—one pre-distorts the offset correlated with each possible N_{R} value and the other combats the slope relevant to ϕ_{CT}.

Fig. 24(a) details the offset calibration. The offset related to each N_{R} value affects D_{TDC} (read in the subsequent FREF cycle) and, thus, can be estimated by accumulating the corresponding D_{TDC}. This is similar to that in [22]. μ_{RT} here is a constant controlling the accumulation speed. By subtracting the estimated offsets, i.e., OS0 ~ OS7, from the fine-tuning path, the effects of the coarse-tuning offsets can be compensated. Prior to the subtraction, the estimated offsets are rounded to the same resolution as ϕ_{CT} by a $\Delta\Sigma$ modulator to avoid the fine resolution of the offsets being masked by the quantization error of the fine-tuning path. Meanwhile, a constant positive phase ϕ_{const} is also added in conjunction with the rounded offset to prevent the fine-tuning path underflow due to the potential negative input. Similar to the $3T_{\mathrm{CKV}}/8$ offset for the metastability mitigation, the extra ϕ_{const} would also shift the Δt_{S} range, without causing functional issues. While the calibration is running, the offset registers would constantly update until the average D_{TDC} corresponding to each N_{R} becomes zero. This indicates that the influences of offsets have been well-compensated, thus becoming invisible to the PLL.

Fig. 24(b) depicts the fine-tuning slope calibration, which detects the slope error by correlating (i.e., accumulating the following product) D_{TDC} with the fine-tuning target ϕ_{CT}, similar to the LMS calibration for K_{DTC} in [8]. μ_{CT} here is a constant controlling the accumulation speed. The correlation output N_{URT} is used to correct the capacitance ratio of C_0/C_{U}, which significantly influences the fine-tuning slope. Instead of directly updating the estimated $\mathbf{E}(C_0/C_{\mathrm{U}})$, which may require long word length and increased hardware cost, we directly tune the physical ratio of C_0/C_{U}: the nominal fixed capacitor C_0 is split into a "real" fixed C_0' and an SC-bank with the unit capacitance of C_{U}'. N_{URT} is dithered by a $\Delta\Sigma$ modulator before adjusting the number of active C_{U}' to tune the "real" capacitance ratio C_0/C_{U} until the slope error vanishes.

Since both calibration paths rely on the same D_{TDC}, they will likely interfere with each other given that both N_{R} and ϕ_{CT} change at a very *slow* rate when the PLL operates in a near-integer channel. This is due to the fact that it is difficult to distinguish the D_{TDC} contribution from the offsets and the slope due to the absence of Δt_{S} dithering mechanism in the overall PLL system, such as that provided by a multi-modulus divider dithered by a high-order $\Delta\Sigma$ modulator. To minimize such mutual interference, the calibration works in the foreground: it is only performed at well-behaved conditions, such as at specific large fractional FCWs. After the calibration is done, the results are frozen and used for nearby channels. The absence of background calibration would not significantly degrade the TAU's performance since it is insensitive to voltage and temperature variations.

VII. Measurement Results

The proposed PLL is fabricated in 40-nm CMOS and occupies an active area of 0.31 mm^2 [excluding output drivers and debugging SRAMs; see Fig. 25(a)]. With a reference clock of 40 MHz, it synthesizes 2.6–4.1 GHz. Fig. 25(b) shows its power breakdown at 2668.2 MHz. The overall PLL consumes 3.48 mW, which is dominated by the DCO and its buffer, costing 2.3 mW at a 1.1-V supply. All other blocks are supplied with 1.0 V. The power consumption for the time mode (e.g., TAU, TDC, and the clock divider for DCO dithering) and digital logic parts is, respectively, 0.65 and 0.52 mW.

Fig. 26(a) shows the measured PN at 2668.2 MHz. The integrated rms jitter (integrated from 10 kHz to 40 MHz, and including all spurs) is 182 fs, almost identical to that in the

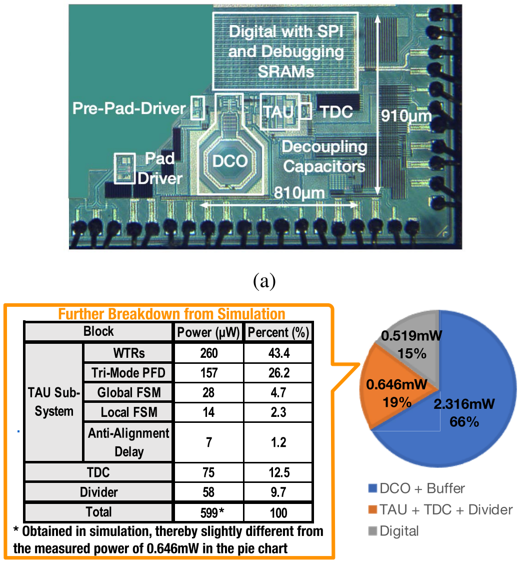
TABLE I

COMPARISON WITH STATE-OF-THE-ART FRACTIONAL-N PLLS

	this work	ISSCC'16 [35]	JSSC'18 [45]	JSSC'20 [43]	JSSC'21 [17]	VLSI'21 [46]	JSSC'21 [42]	JSSC'22 [47]	ISSCC'21 [48]
Process (nm)	**40**	28	65	28	130	65	14	28	65
Phase detection strategy	**TAU + TDC**	DTC + SPD[1]	DTC + TA + TDC	DTC + BBPD[2]	Voltage domain	DTC + SPD[1]	DTC + SPD[1]	DTC + BBPD[2]	Voltage domain
Ref. freq. (MHz)	**40**	40	26 × 2	500	80	50	76.8 × 2	250	150
Osc. freq. (GHz)	**2.68**	3.88	2.44	13.5	3.36	3.3	6.2	13	15
Int. rms jitter (fs)	**182**	159	535	66.2	101	263	96.3	99.6	104
Worst frac. spur (dBc)	**-59**	-57.5[3]	-56	-61	-56	-53	-68[3]	-51.1	-61
Ref. spur (dBc)	**-73.5**	-81.5[3]	-72	-80.1	-79	-80	-63.6[3]	-73.2	NA
Built-in resilience to supply and temperature	**Yes**	No	No	No	Yes	No	No	No	No
Power (mW)	**3.5**	8.2	0.98	19.8	9.2	4.6	8.2	10.8	7.3
FoM[4] (dB)	**-249.4**	-246.8	-246	-250.6	-250.3	-246	-251.2	-249.7	-251
$\mathrm{FoM_N}$[5] (dB)	**-267.7**	-266.7	-265.7	-264.9	-266.5	-264.2	-270.3	-266.9	-271
Active area (mm^2)	**0.31**	0.3	0.23	0.17	0.27	0.48	0.31	0.21	0.21

[1] Sampling phase detector [2] Bang-bang phase detector [3] Normalized to osc. frequency

[4] $\mathrm{FoM} = 10 \cdot \log_{10}[(\mathrm{jitter}/1\,\mathrm{s})^2 \cdot \mathrm{power}/1\,\mathrm{mW}]$ [5] $\mathrm{FoM_N} = 10 \cdot \log_{10}[(\mathrm{jitter}/1\,\mathrm{s})^2 \cdot \mathrm{power}/1\,\mathrm{mW}/(\mathrm{osc.freq.}/\mathrm{ref.freq.})]$



(a)

Further Breakdown from Simulation

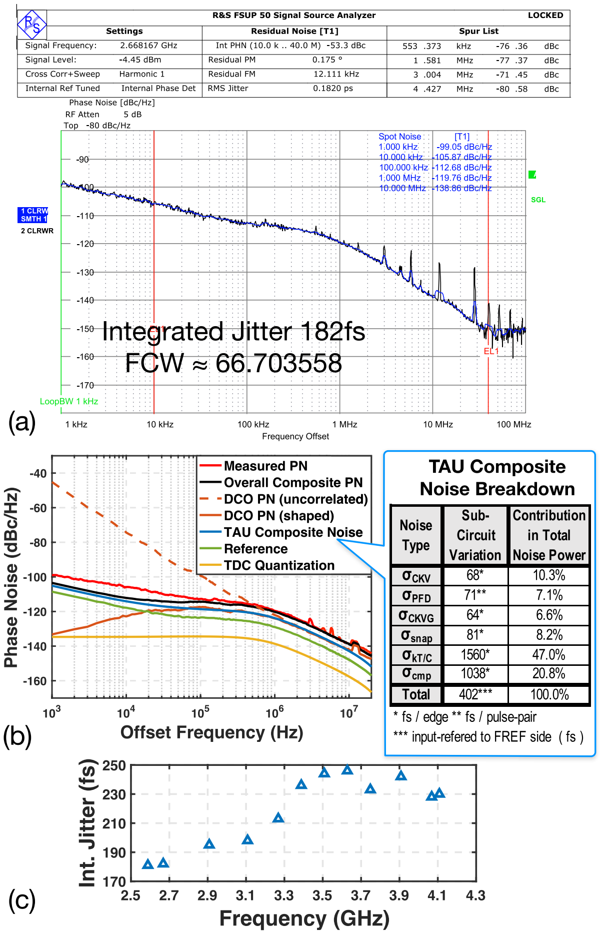
Block		Power (µW)	Percent (%)
TAU Sub-System	WTRs	260	43.4
	Tri-Mode PFD	157	26.2
	Global FSM	28	4.7
	Local FSM	14	2.3
	Anti-Alignment Delay	7	1.2
TDC		75	12.5
Divider		58	9.7
Total		599*	100

* Obtained in simulation, thereby slightly different from the measured power of 0.646mW in the pie chart

(b)

Fig. 25. (a) Chip micrograph. (b) Power consumption breakdown.

nearby integer-N channel (177 fs at 2640 MHz). Considering the total power consumption of 3.48 mW, this PLL achieves a jitter-power FoM [49] of −249.4 dB. Fig. 26(b) compares the measured PN with its s-domain prediction, indicating a tight agreement at offset frequencies above 50 kHz. In this s-domain model, the input referred-jitter of TAU is 402 fs, estimated by simulating the jitter of each sub-circuit and combining the contributors via (30). The corresponding contribution to PN is obtained by an amended formula to (31) that combines the sub-block's noise in the spectrum domain. The noise contribution of each sub-circuit is also listed in Fig. 26(b). Fig. 26(c) shows the integrated rms jitter across frequencies with the same fractional FCW as 2668.2 MHz, i.e., $\mathrm{FCW_{frac}} \approx 0.7$. The measured jitter degrades as the frequency increases. We suspect the dramatic degradation between 3300 and 3800 MHz is attributed to the nearby inductors in this SoC and unoptimized implementation of the DCO SC tuning banks to support the wideband direct phase modulation [36].


R&S FSUP 50 Signal Source Analyzer LOCKED

Settings		Residual Noise [T1]		Spur List			
Signal Frequency:	2.668167 GHz	Int PHN (10.0 k .. 40.0 M)	-53.3 dBc	553 .373	kHz	-76 .36	dBc
Signal Level:	-4.45 dBm	Residual PM	0.175 °	1 .581	MHz	-77 .37	dBc
Cross Corr+Sweep	Harmonic 1	Residual FM	12.111 kHz	3 .004	MHz	-71 .45	dBc
Internal Ref Tuned	Internal Phase Det	RMS Jitter	0.1820 ps	4 .427	MHz	-80 .58	dBc

Phase Noise [dBc/Hz]
RF Atten 5 dB
Top -80 dBc/Hz

Spot Noise	[T1]
1.000 kHz	-99.05 dBc/Hz
10.000 kHz	-105.87 dBc/Hz
100.000 kHz	-112.68 dBc/Hz
1.000 MHz	-119.76 dBc/Hz
10.000 MHz	-138.86 dBc/Hz

TAU Composite Noise Breakdown

Noise Type	Sub-Circuit Variation	Contribution in Total Noise Power
σ_{CKV}	68*	10.3%
σ_{PFD}	71**	7.1%
σ_{CKVG}	64*	6.6%
σ_{snap}	81*	8.2%
$\sigma_{\mathrm{kT/C}}$	1560*	47.0%
σ_{cmp}	1038*	20.8%
Total	402***	100.0%

* fs / edge ** fs / pulse-pair

*** input-refered to FREF side (fs)

Fig. 26. (a) Measured PN at 2668.2 MHz. (b) Comparison between the measured PN in (a) and its s-domain model prediction. In the jitter breakdown table, $\sigma_{\mathrm{KT/C}}$ is estimated with C_0 of 1.6 pF and discharge slope of 33.8 μV/ps; others are obtained by simulation. These jitter contributions are combined as per (30) to estimate the TAU composite noise. (c) RMS jitter (integrated from 10 kHz to 40 MHz) across carrier frequencies with fractional FCW ($\mathrm{FCW_{frac}}$) of 0.7.

To demonstrate the TAU's advantages in suppressing fractional spurs, the PLL output spectrum is measured in a near-integer channel of 2680.04 MHz (FCW ≈ 67.00025).

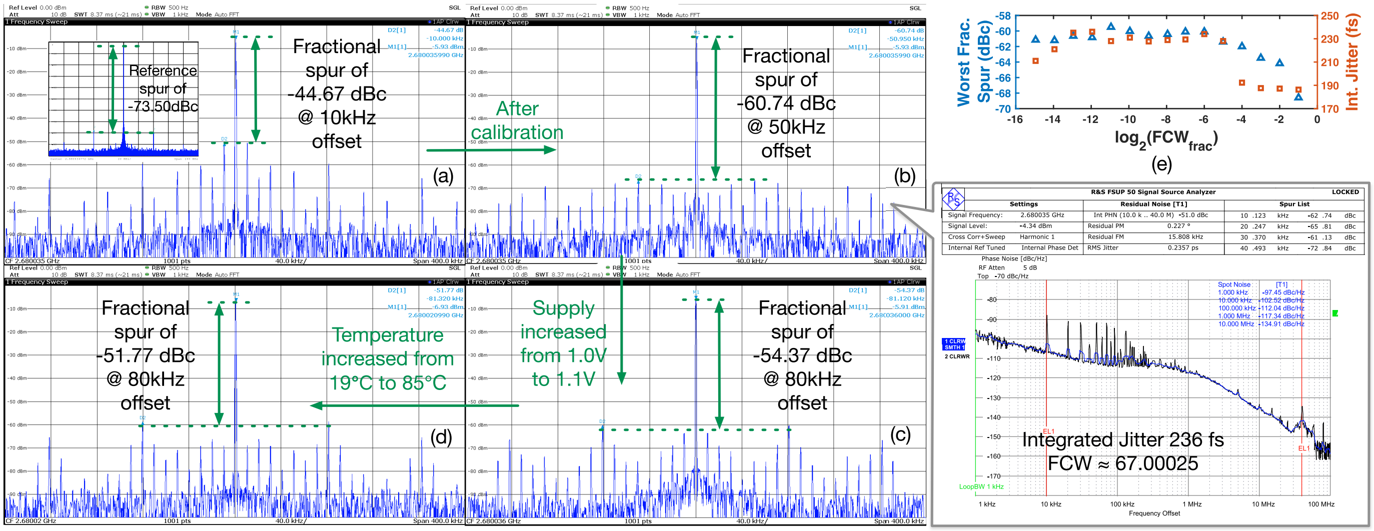


Fig. 27. (a)–(d) Measured spectra at 2680.04 MHz (FCW ≈ 67.00025): (a) baseline (without calibration) and (b) with calibration. From (b) to (d), the supply and temperature change incrementally with the calibration code frozen. (e) Worst case fractional spur level and the corresponding integrated rms jitter versus fractional FCW (FCW_{frac}) with integer FCW fixed at 67.

According to Fig. 27(a), the worst-case fractional spur is −44.67 dBc. Note that they are measured *before* any TAU calibration, e.g., for global gain and INL. This compares favorably with the literature reports of worst case fractional spurs in DTC-based PLLs that adopt only gain calibration but with no further DTC linearity enhancement techniques, e.g., −37 dBc in [20] and −42 dBc in [8]. Our fundamental design choice—adopting T_{CKV}, the PLL carrier period, as the basis for the time offset cancellation—is thus validated. This "golden" base automatically scales the global gain of the TAU transfer function, thus avoiding any need for the corresponding calibration.

The fractional spurs in Fig. 27(a) are dominated by the TAU's INL, chiefly due to the coarse-tuning non-ideality and the gain error in fine-tuning. After compensating the INL with the piecewise calibration, the worst case fractional spur becomes −60.74 dBc @50 kHz, the fifth fractional spur in Fig. 27(b). In this scenario, the integrated rms jitter is 236 fs. The worst-case fractional spur levels and integrated rms jitter are swept for at the fractional channels close to 2680 MHz. As shown in Fig. 27(e), all the spur levels are below −59 dBc.

Since the TAU utilizes the time basis of T_{CKV}, which is constantly tracked by the PLL, the TAU-based PLL is expected to exhibit inherent resilience to environmental changes, i.e., supply and temperature drifts. To prove this, we froze the TAU's INL calibration setting and then measured the spur levels under certain environmental changes. From Fig. 27(b) to (c), the TAU's supply was increased from 1.0 to 1.1 V, and the worst spur remains −54.37 dBc. From Fig. 27(c) to (d), the environment temperature was increased from 19 °C to 85 °C, and the worst spur level is still below −51.7 dBc. These are noteworthy improvements compared with the DTC-based counterparts, as they would generate substantial spurs if their transfer function drift could not be compensated. For example, Chen *et al.* [41] reported a 14% DTC resolution drift when its supply increased from 1.0 to 1.1 V. As measured in [20], a 10% DTC gain error can cause an in-band fractional spur higher than −30 dBc.

Table I summarizes and compares the performance of the proposed PLL with the state-of-the-art fractional-N PLLs. This work achieves the competitive spur level below −59 dBc and a state-of-the-art tradeoff between jitter and power, i.e., FoM of −249.4 under the low power constraint.

VIII. Conclusion

This article introduces a fractional-N PLL based on the proposed TAU, which extracts the phase error by calculating a weighted sum of its time-domain inputs derived from timestamps of the reference and DCO clocks. The prototype PLL demonstrates low-spur levels, which are robust under supply and temperature drift. Such spurious performance benefits from the phase-error-extraction strategy—scaling the "golden" time base, i.e., DCO period, to cancel the PD input—which automatically corrects the TAU's transfer function. The methodology-level improvement indicates a potential for exploring this new phase-detection category for low-spur clock generation.

Acknowledgment

The authors wish to thank Ken Yamamoto of Sony Semiconductor Solution for sponsoring this research and many technical discussions, and Atef Akhnoukh and Zu Yao Chang of the Delft University of Technology for technical support.

References

[1] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A low noise sub-sampling PLL in which divider noise is eliminated and PD/CP noise is not multiplied by N^2," *IEEE J. Solid-State Circuits*, vol. 44, no. 12, pp. 3253–3263, Dec. 2009.

[2] X. Gao, E. A. M. Klumperink, G. Socci, M. Bohsali, and B. Nauta, "Spur reduction techniques for phase-locked loops exploiting a sub-sampling phase detector," *IEEE J. Solid-State Circuits*, vol. 45, no. 9, pp. 1809–1821, Sep. 2010.

[3] A. Sharkia, S. Mirabbasi, and S. Shekhar, "A 0.01 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.

[4] Z. Yang, Y. Chen, S. Yang, P.-I. Mak, and R. P. Martins, "16.8 a 25.4-to-29.5 GHz 10.2 mW isolated sub-sampling PLL achieving -252.9 dB jitter-power FoM and −63dBc reference spur," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 270–272.

[5] Z. Yang, Y. Chen, J. Yuan, P.-I. Mak, and R. P. Martins, "A 3.3-GHz integer N-type-II sub-sampling PLL using a BFSK-suppressed push–pull SS-PD and a fast-locking FLL achieving -82.2-dBc REF spur and −255-dB FOM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 2, pp. 238–242, Feb. 2022.

[6] H. Liu *et al.*, "A 265-μW fractional-N digital PLL with seamless automatic switching sub-sampling/sampling feedback path and duty-cycled frequency-locked loop in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 3478–3492, Dec. 2019.

[7] W. Wu *et al.*, "A 28-nm 75-fs$_{rms}$ 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.

[8] 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-fs$_{rms}$ integrated jitter at 4.5-mW power," *IEEE J. Solid-State Circuits*, vol. 46, no. 12, pp. 2745–2758, Dec. 2011.

[9] J. Zhuang and R. Staszewski, "A low-power all-digital PLL architecture based on phase prediction," in *Proc. IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2012, pp. 797–800.

[10] S. Levantino, G. Marzin, and C. Samori, "An adaptive pre-distortion technique to mitigate the DTC nonlinearity in digital PLLs," *IEEE J. Solid-State Circuits*, vol. 49, no. 8, pp. 1762–1772, Aug. 2014.

[11] A. Elkholy, T. Anand, W.-S. Choi, A. Elshazly, and P. K. Hanumolu, "A 3.7 mW low-noise wide-bandwidth 4.5 GHz digital fractional-N PLL using time amplifier-based TDC," in *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 867–881, Apr. 2015.

[12] H. Liu *et al.*, "A 0.98 mW fractional-N ADPLL using 10b isolated constant-slope DTC with FOM of −246 dB for IoT applications in 65 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 246–248.

[13] B. Liu *et al.*, "A 1.2 ps-jitter fully-synthesizable fully-calibrated fractional-N injection-locked PLL using true arbitrary nonlinearity calibration technique," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–4.

[14] V. K. Chillara *et al.*, "9.8 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," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 172–173.

[15] R. B. Staszewski and P. T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*, 1st ed. Hoboken, NJ, USA: Wiley, Sep. 2006.

[16] S. Levatino, G. Marucci, G. Marzin, A. Fenaroli, C. Samori, and A. L. Lacaita, "A 1.7 GHz fractional-N frequency synthesizer based on a multiplying delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2678–2691, Nov. 2015.

[17] L. Wu, T. Burger, P. Schönle, and Q. Huang, "A power-efficient fractional-N DPLL with phase error quantized in fully differential-voltage domain," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1254–1264, Apr. 2021.

[18] J. Tao and C.-H. Heng, "A 2.2-GHz 3.2-mW DTC-free sampling $\Delta\Sigma$ fractional-N PLL with −110-dBc/Hz in-band phase noise and −246-dB FoM and −83-dBc reference spur," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3317–3329, Sep. 2019.

[19] Y. Chen, J. Gong, R. B. Staszewski, and M. Babaie, "A fractional-N digitally intensive PLL achieving 428-fs jitter and <-54-dBc spurs under 50-mV$_{pp}$ supply ripple," *IEEE J. Solid-State Circuits*, vol. 57, no. 6, pp. 1749–1764, Jun. 2021.

[20] Y.-H. Liu *et al.*, "An ultra-low power 1.7–2.7 GHz fractional-N sub-sampling digital frequency synthesizer and modulator for IoT applications in 40 nm CMOS," in *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 5, pp. 1094–1105, May 2017.

[21] K.-F. Un *et al.*, "A 0.12-mm^2 1.2-to-2.4-mW 1.3-to-2.65-GHz fractional-N bang-bang digital PLL with 8-μs settling time for multi-ISM-band ULP radios," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3307–3316, Sep. 2019.

[22] N. Markulic *et al.*, "A DTC-based subsampling PLL capable of self-calibrated fractional synthesis and two-point modulation," *IEEE J. Solid-State Circuits*, vol. 51, no. 12, pp. 3078–3092, Dec. 2016.

[23] W. Wu *et al.*, "32.2 a 14 nm analog sampling fractional-N PLL with a digital-to-time converter range-reduction technique achieving 80 fs integrated jitter and 93 fs at near-integer channels," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 444–446.

[24] T. Seong *et al.*, "17.3 A −58 dBc-worst-fractional-spur and −234 dB-FoM$_{jitter}$, 5.5 GHz ring-DCO-based fractional-N DPLL using a time-invariant-probability modulator, generating a nonlinearity-robust DTC-control word," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 270–272.

[25] D. Liao and F. F. Dai, "A fractional-N reference sampling PLL with linear sampler and CDAC based fractional spur cancellation," *IEEE J. Solid-State Circuits*, vol. 56, no. 3, pp. 694–704, Mar. 2021.

[26] J. Z. Ru, C. Palattella, P. Geraedts, E. Klumperink, and B. Nauta, "A high-linearity digital-to-time converter technique: Constant-slope charging," *IEEE J. Solid-State Circuits*, vol. 50, no. 6, pp. 1412–1423, Jun. 2015.

[27] A. Tharayil Narayanan *et al.*, "A fractional-N sub-sampling PLL using a pipelined phase-interpolator with an FoM of -250 dB," *IEEE J. Solid-State Circuits*, vol. 51, no. 7, pp. 1630–1640, Jul. 2016.

[28] F.-W. Kuo *et al.*, "An all-digital PLL for cellular mobile phones in 28-nm CMOS with −55 dBc fractional and −91 dBc reference spurs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3756–3768, Nov. 2018.

[29] Z. Gao *et al.*, "A 2.6-to-4.1 GHz fractional-N digital PLL based on a time-mode arithmetic unit achieving −249.4 dB FoM and -59 dBc fractional spurs," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 380–382.

[30] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm CMOS using time-register," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, Apr. 2014.

[31] Y. Wu, P. Lu, and R. B. Staszewski, "A time-domain 147fs$_{rms}$ 2.5-MHz bandwidth two-step flash-MASH 1-1-1 time-to-digital converter with third-order noise-shaping and mismatch correction," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 8, pp. 2532–2545, Aug. 2020.

[32] P. Effendrik, W. Jiang, M. Van De Gevel, F. Verwaal, and R. B. Staszewski, "Time-to-digital converter (TDC) for Wimax ADPLL in 40-nm CMOS," in *Proc. 20th Eur. Conf. Circuit Theory Design (ECCTD)*, Aug. 2011, pp. 365–368.

[33] P. J. A. Harpe *et al.*, "A 26μW 8 bit 10 MS/s asynchronous SAR ADC for low energy radios," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1585–1595, Jul. 2011.

[34] M. Zhang, C.-H. Chan, Y. Zhu, and R. P. Martins, "A 0.6-V 13-bit 20-MS/s two-step TDC-assisted SAR ADC with PVT tracking and speed-enhanced techniques," *IEEE J. Solid-State Circuits*, vol. 54, no. 12, pp. 3396–3409, Dec. 2019.

[35] X. Gao *et al.*, "9.6 A 2.7-to-4.3 GHz, 0.16 psrms-jitter, −246.8 dB-FOM, digital fractional-N sampling PLL in 28nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 174–175.

[36] Z. Gao *et al.*, "A DPLL-based phase modulator achieving −46 dB EVM with a fast two-step DCO nonlinearity calibration and non-uniform clock compensation," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2022, pp. 14–15.

[37] A. Homayoun and B. Razavi, "Analysis of phase noise in phase/frequency detectors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 3, pp. 529–539, Mar. 2013.

[38] R. B. Staszewski, C. Fernando, and P. T. Balsara, "Event-driven simulation and modeling of phase noise of an RF oscillator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 4, pp. 723–733, Apr. 2005.

[39] A. A. Abidi, "Phase noise and jitter in CMOS ring oscillators," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1816, Aug. 2006.

[40] L. Vercesi, L. Fanori, F. D. Bernardinis, A. Liscidini, and R. Castello, "A dither-less all digital PLL for cellular transmitters," *IEEE J. Solid-State Circuits*, vol. 47, no. 8, pp. 1908–1920, Aug. 2012.

[41] P. Chen, F. Zhang, Z. Zong, S. Hu, T. Siriburanon, and R. B. Staszewski, "A 31-μW, 148-fs step, 9-bit capacitor-DAC-based constant-slope digital-to-time converter in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 11, pp. 3075–3085, Nov. 2019.

[42] 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.

[43] A. Santiccioli *et al.*, "A 66-fs-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.

[44] B. Liu *et al.*, "A fully synthesizable Fractional-N MDLL with zero-order interpolation-based DTC nonlinearity calibration and two-step hybrid phase offset calibration," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 603–616, Feb. 2021.

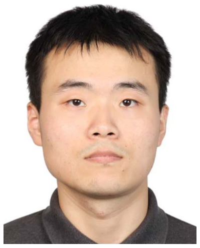
[45] H. Liu, D. Tang, Z. Sun, W. Deng, H. C. Ngo, and K. Okada, "A sub-mW fractional-*N* ADPLL With FOM of −246 dB for IoT applications," *IEEE J. Solid-State Circuits*, vol. 53, no. 12, pp. 3540–3552, Dec. 2018.

[46] Z. Xu, M. Osada, and T. Iizuka, "A 3.3-GHz 4.6-mW fractional-N type-II hybrid switched-capacitor sampling PLL using CDAC-embedded digital integral path with −80-dBc reference spur," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[47] 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.

[48] J. Kim *et al.*, "32.4 A 104fs_{rms}-jitter and −61 dBc-fractional spur 15 GHz fractional-N subsampling PLL using a voltage-domain quantization-error cancelation technique," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2021, pp. 448–450.

[49] X. Gao, E. A. M. Klumperink, P. F. J. Geraedts, and B. Nauta, "Jitter analysis and a benchmarking figure-of-merit for phase-locked loops," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 2, pp. 117–121, Feb. 2009.



Zhong Gao (Graduate Student Member, IEEE) received the B.Sc. degree in physics from Shandong University, Jinan, China, in 2011, and the M.Sc. degree in microelectronics from the University of Chinese Academy of Sciences, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in electronic engineering with the Department of Microelectronics, Delft University of Technology (TU Delft), Delft, The Netherlands.

Before joining TU Delft, he worked on wireless transceiver design at Altobeam Inc., Beijing. His current research interests include mixed-signal IC and RF transceiver system design.

Jingchu He received the B.Sc. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008, and the M.Sc. degree in microelectronics from the Delft University of Technology, Delft, The Netherlands, in 2011.

From 2019 to 2022, she was an RFIC Designer with the Delft University of Technology. She is currently a System Application Engineer with the Automotive Section, NXP Semiconductor, Eindhoven, The Netherlands.

Martin Fritz received the Diploma degree in electrical engineering from the University of Stuttgart, Stuttgart, Germany, in 2009.

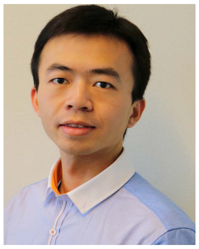
Since 2010, he has been an Engineer with the Sony Stuttgart Technology Center, Stuttgart. His main research and development interests include wireless communications and sensing systems, and related digital signal processing and system design.

Jiang Gong (Student Member, IEEE) received the B.Sc. degree in electrical engineering from Jilin University, Changchun, China, in 2015, and the M.Sc. degree *(cum laude)* in microelectronics from the Delft University of Technology, Delft, The Netherlands, in 2017, where he is currently pursuing the Ph.D. degree.

His research focuses on high-spectral-purity and wide-tuning-range cryogenic frequency synthesizers for quantum computing applications.

Yiyu Shen (Member, IEEE) received the M.S. degree in microelectronics from Tsinghua University, Beijing, China, and Katholieke Universiteit Leuven, Leuven, Belgium, in 2014, and the Ph.D. degree in electrical engineering from the Delft University Technology, Delft, The Netherlands, in 2021.

His current research interests include power amplifiers and digital-assisted RF integrated circuits and systems.



Zhirui Zong (Member, IEEE) received the B.Eng. degree (Hons.) in electronic and information engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012, and the Ph.D. degree in microelectronics from the Delft University of Technology, Delft, The Netherlands, in 2019.

From 2018 to 2021, he has been a Senior Analog and RF IC designer with NXP Semiconductors, Eindhoven, The Netherlands. He is currently an Assistant Professor with The Hong Kong University of Science and Technology, Guangzhou, China. His current research interests include frequency synthesis techniques, RF and mixed-signal integrated circuits for wireless/wireline communications, and radar systems.

Peng Chen (Member, IEEE) received the B.Sc. degree in electronics from the Huazhong University of Science and Technology, Wuhan, China, in 2012, the M.Sc. degree in microelectronic from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 2014, and the Ph.D. degree from University College Dublin, Dublin, Ireland, in 2019.

From 2017 to 2018, he was a Visiting Research Assistant with the University of Macau, Macau. From 2019 to 2021, he was a Post-Doctoral Researcher with Lund University, Lund, Sweden. He is currently with Wuxi Grandmicro, Wuxi, China. His research interests include time-domain data converters and frequency synthesizers.

Gerd Spalink received the Diploma degree (Hons.) in communications engineering from RWTH Aachen University, Aachen, Germany, in 1990.

He joined the Research Department, Sony Stuttgart Technology Center, Stuttgart, Germany, in 1990. His recent work areas at Sony Europe B.V., Stuttgart, Germany, are silicon tuners for digital and analog broadcast reception and Bluetooth low-energy transceivers. He has contributed digital assistance circuits to several large-scale integration (LSI) designs with a focus on low power consumption.

Ben Eitel received the Diploma and Ph.D. degrees in electrical engineering from the University of Stuttgart, Stuttgart, Germany, in 2000 and 2009, respectively.

Since 2006, he has been with the Sony Stuttgart Technology Center, Stuttgart. He is currently the Manager of the RF Communications Group, Sony Europe B.V., Stuttgart. His main research and development interests include RF communications and sensing solutions.

Morteza S. Alavi (Member, IEEE) received the B.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 2003, the M.Sc. degree in electrical engineering from the University of Tehran, Tehran, in 2006, and the Ph.D. degree in electrical engineering from the Delft University of Technology (TU-Delft), Delft, The Netherlands, in 2014.

He was a Co-Founder and the CEO of DitIQ B.V., Delft, a local company developing energy-efficient, wideband wireless transmitters for the next generation of the cellular network. In September 2016, he joined the ELCA Research Group, TU-Delft, as a Faculty Member, where he is currently a tenured Assistant Professor. He has coauthored a book *Radio-Frequency Digital-to-Analog Converter* (Elsevier, 2016). His main research interests are designing high-frequency and high-speed wireless/cellular communication and sensor systems, as well as in the field of wireline transceivers.

Dr. Alavi was the Best Paper Award recipient of the 2011 IEEE International Symposium on Radio-Frequency Integrated Technology (RFIT). He received the Best Student Paper Award (Second Place) at the 2013 Radio-Frequency Integrated Circuits (RFIC) Symposium. His Ph.D. student also won the Best Student Paper Award (First Place) at the 2017 RFIC Symposium held in Honolulu, HI, USA. One of his students recently received the 2021 Institute of Semiconductor Engineers (ISE) President Best Paper Award of the International SoC Design Conference (ISOCC).

Robert Bogdan Staszewski (Fellow, IEEE) was born in Bialystok, Poland. He received the B.Sc. *(summa cum laude)*, M.Sc., and Ph.D. degrees in electrical engineering from The University of Texas at Dallas, Richardson, TX, USA, in 1991, 1992, and 2002, respectively.

From 1991 to 1995, he was with Alcatel Network Systems, Richardson, TX, involved in SONET cross-connect systems for fiber optics communications. He joined Texas Instruments Inc., Dallas, TX, USA, in 1995, where he was elected as a Distinguished Member of Technical Staff (limited to 2% of technical staff). From 1995 to 1999, he was engaged in advanced CMOS read channel development for hard disk drives. In 1999, he co-started the Digital RF Processor (DRP) Group, Texas Instruments Inc., with a mission to invent new digitally intensive approaches to traditional RF functions for integrated radios in deeply scaled CMOS technology. He was the CTO of the DRP Group from 2007 to 2009. In 2009, he joined the Delft University of Technology, Delft, The Netherlands, where he currently holds a guest appointment of Full Professor (*Antoni van Leeuwenhoek Hoogleraar*). Since 2014, he has been a Full Professor with the University College Dublin (UCD), Dublin, Ireland. He is also a Co-Founder of a startup company, Equal1 Labs, Dublin, with design centers located in Silicon Valley, CA, USA, and Dublin, aiming to produce single-chip CMOS quantum computers. He has authored or coauthored seven books, eight book chapters, and 160 journal and 220 conference publications. He holds 210 issued U.S. patents. His research interests include nanoscale CMOS architectures and circuits for frequency synthesizers, transmitters, and receivers, as well as quantum computers.

Prof. Staszewski was a recipient of the 2012 IEEE Circuits and Systems Industrial Pioneer Award.

Masoud Babaie (Senior Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2004, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, in 2006, and the Ph.D. degree *(cum laude)* in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 2016.

From 2006 to 2011, he was with the Kavoshcom Research and Development Group, Tehran, where he was involved in designing wireless communication systems. From 2014 to 2015, he was a Visiting Scholar Researcher with the Berkeley Wireless Research Center, Berkeley, CA, USA. In 2016, he joined the Delft University of Technology, where he is currently an Associate Professor. He has authored or coauthored one book, three book chapters, 11 patents, and more than 80 technical articles. His research interests include RF/millimeter-wave integrated circuits and systems for wireless communications and cryogenic electronics for quantum computation.

Dr. Babaie was a co-recipient of the 2015–2016 IEEE Solid-State Circuits Society Pre-Doctoral Achievement Award, the 2019 IEEE ISSCC Demonstration Session Certificate of Recognition, the 2020 IEEE ISSCC Jan Van Vessem Award for Outstanding European Paper, and the 2022 IEEE CICC Best Paper Award. He was a recipient of the Veni Award from the Netherlands Organization for Scientific Research (NWO) in 2019. He also serves on the Technical Program Committee of the IEEE International Solid-State Circuits Conference (ISSCC). He also serves as the Co-Chair of the Emerging Computing Devices and Circuits Subcommittee of the IEEE European Solid-State Circuits Conference (ESSCIRC).