UNIVERSITY OF CALIFORNIA SAN DIEGO

Enhancement Techniques for Digital Phase-Locked Loops

A dissertation submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy

in

Electrical Engineering (Electronic Circuits and Systems)

by

Amr Ibrahim Farag Eissa

Committee in charge:

Professor Ian Galton, Chair
Professor Peter Asbeck
Professor Gert Cauwenberghs
Professor Truong Nguyen

2024

The dissertation of Amr Ibrahim Farag Eissa is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2024

DEDICATION

To My Late Father

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

All praise and gratitude are due to ALLAH. I dedicate this thesis to my father, who passed away two years ago. It is beyond comprehension how much I love and miss him. He is, unquestionably, my role model, and I will be pleased if I can follow in his footsteps in all aspects of life. No words can express how grateful I am for all what he taught me, and all the sacrifices he made for me and my family. His love, encouragement, advice, wisdom, strength, ethics, kindness, and support were invaluable to me throughout my life. Earning my PhD degree is all what he wished for. I'm sure that he would have been very happy and proud at this moment had he been with us. Thank you … for everything.

I would like to thank Prof. Ian Galton for his generous financial support and meticulous technical supervision throughout my PhD studies. I thank him for his patience and understanding of the personal issues that I had to deal with over the course of the last few years. It would take pages to share the lessons I learned from him, but I would quote "never assume that what you hear or read is correct, think deeply about it first and try to come up with arguments to falsify it". Also, as ISPG graduates, we are lucky to inherit his academic genealogy and have David Hilbert as our academic fourth great-grandparent!!

I am grateful to have Prof. Peter Asbeck, Prof. Gert Cauwenberghs, and Prof. Truong Nguyen on my committee. My special thanks to Prof. Asbeck for recommending me to join the PhD program at UCSD. I was lucky to enroll in the last offering of his famous ECE265C course, before he retired, and learn about the world of Power Amplifiers.

I would like to thank Colin W. Wu, Eric Fogleman, Yiwu Tang, and Dongmin Park for their technical guidance and selfless sharing of their industry experience throughout my studies.

Special thanks to Colin W. Wu for setting up the simulation platform and providing the SystemVerilog files used to generate the behavioral simulations in Chapter 4.

Next, the three heroines and two kids. First, my mother. I genuinely don't think motherhood can get any better. Her love, support, kind words, and strength are beyond imagination. In addition to the many things she did for our family, I owe her for helping with my studies since I was a kid, especially mathematics. During my stay in the US, my sister took care of my mother and father, filling the gap I left and more. I don't think I'll ever be able to thank her enough for that. Her support and love helped me overcome many of the challenges I had during my studies. Finally, my wife. She made a lot of sacrifices and stood with me against many challenges, especially during my PhD studies. She selflessly dedicated her life to our kids and to making us all happy. I always remind her that she shares this PhD title with me as many of the hours I spent working were only possible because she chose to do what I should have otherwise done. To my two beloved kids, thank you for giving meaning to what I am doing. I love you so much and will always be there for you.

I am in debt to Prof. Mohamed El-Nozahi, my master's advisor, who played a key role in shaping my career. I am grateful to have Dr. Samer Idres as a friend and mentor. Aside of his support, he never failed to recommend the finest dining places in Egypt and the US.

Moving to the ISPG members, I am grateful to get to know Enrique on my first day in the group. I enjoyed every moment in the countably infinite discussions we had. His critical reasoning and logical thinking process sharpened mine (so blame him if you find them messed up). He is a leader, always supportive and never fails to provide practical advice. I met Raghu next. I admire his knowledge and I enjoyed the discussions we had. Unfortunately, most of

them ended up with me realizing that I still must learn a lot to catch up with what he was saying, but I'm grateful for that. He indirectly taught me how to better understand technical material, and I benefited from his wise advice. He helped me deal with too many software-related issues, and his presence in the lab was certainly a privilege for all other lab members. Next comes Eslam whose hard work is a role model. He selflessly shared many of the industrial lessons he learnt with me, and discussions with him made me a better designer. I thank former lab members Mohamed Badr and the optimizer (a.k.a. Ahmed Hussein) for the discussions we had and the quality time we spent together. I wish both the best of luck in their current and future endeavors.

Finally, I would like to thank the Egyptian Mafia at UCSD (too many names to list :)). Special thanks to Mohammed Helal for his encouragement and support.

Chapter 1, in part, is currently being prepared for submission for publication of the material. A. I. Eissa, E. Alvarez-Fontecilla, E. Helal, I. Galton. The dissertation author is the primary investigator and author of this material. Professor Ian Galton supervised the research which forms the basis for this material.

Chapter 2, in full, is under review in the IEEE Transactions on Circuits and Systems I: Regular Papers. A. I. Eissa, E. Alvarez-Fontecilla, C. Weltin-Wu, I. Galton, 2024. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

Chapter 3, in part, is currently being prepared for submission for publication of the material. A. I. Eissa, I. Galton. The dissertation author is the primary investigator and author of this material. Professor Ian Galton supervised the research which forms the basis for this material.

VITA

2024 Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems), University of California San Diego, USA.

2016 Master of Science in Electrical Engineering, Ain Shams University, Cairo, Egypt.

2012 Bachelor of Science in Electrical Engineering, Ain Shams University, Cairo, Egypt.

ABSTRACT OF THE DISSERTATION

Enhancement Techniques for Digital Phase-Locked Loops

by

Amr Ibrahim Farag Eissa

Doctor of Philosophy in Electrical Engineering (Electronic Circuits and Systems)

University of California San Diego, 2024

Professor Ian A. Galton, Chair

The performance of phase-locked loops (PLLs) is critical to advancing the data rates in wired and wireless communication systems. Most PLLs incorporate either analog filters and voltage-controlled oscillators (VCOs) or digital filters and digitally-controlled oscillators (DCOs). The former are called analog PLLs and the latter are called digital PLLs. To date, analog PLLs have the best phase error performance, but digital PLLs occupy smaller active area, lend themselves better to digital calibration and signal processing techniques, and are more

compatible with highly-scaled CMOS integrated circuit (IC) technology. Thus, improving the performance of digital PLLs has been the subject of intensive research for many years.

The first chapter of this dissertation presents an incremental frequency control (IFC) scheme for DCOs comprised of an arbitrarily large bank of unit-weighted frequency control elements (FCEs). The scheme requires only a pair of differential 1-bit control signals, is inherently monotonic, and avoids transient frequency glitches. Measurement results are presented to demonstrate the functionality of the proposed frequency control scheme and its negligible impact on a PLL's locking time and phase noise.

The second chapter of this dissertation presents a reference frequency-doubling (RFD) technique that is immune to crystal oscillator duty-cycle error and is not subject to the speed-accuracy trade-off associated with conventional duty-cycle error calibration techniques. The technique is presented and analyzed in the context of a delta-sigma frequency-to-digital converter (ΔΣ-FDC) based PLL. Analysis and behavioral simulations with nonideal circuit parameters show a 10× improvement in the worst-case convergence time compared to prior art.

The third chapter of this dissertation describes a parasitic-capacitance-induced nonlinearity mechanism in charge pumps (CPs) used in fractional-*N* PLLs, along with a scheme to mitigate it. Presented in the context of a 10 GHz ΔΣ-FDC based PLL, behavioral simulations with nonideal circuit parameters show that the proposed technique reduces the PLL's fractional spurs' level by more than 10 dB, achieving a worst-case in-band spur level below −54 dBc and an integrated RMS jitter below 80 fs.

The fourth chapter of this dissertation presents a system architecture review, along with behavioral simulation results, for a 9–11 GHz ΔΣ-FDC PLL IC, targeting 75 fs_{rms} jitter.

CHAPTER 1

AN INCREMENTAL FREQUENCY CONTROL SCHEME FOR DIGITALLY CONTROLLED OSCILLATORS

Abstract—The frequency of a digitally-controlled oscillator (DCO) is typically adjusted by changing the state of one or more of its frequency-control elements (FCEs), at specific times, via digital control signals. FCE banks comprised of power-of-two-weighted FCEs are attractive as they require small numbers of control signals, but they are sensitive to component and control signal propagation-delay mismatches that introduce nonlinearity and frequency glitches. Unit-weighted FCE banks are less prone to these issues. However, they require a relatively large number of control signals even when using row-column control schemes, in part, because redundant control signals are needed to avoid timing mismatches. This letter presents and experimentally demonstrates an incremental frequency control (IFC) scheme to control arbitrarily large banks of unit-weighted FCEs. The scheme requires only a pair of differential 1-bit control signals, and it avoids frequency glitches.

I. INTRODUCTION

Digital phase-locked loops (PLLs) have become increasingly popular over the last few decades [1], [2], [3], [4], [5], [6]. At the core of a digital PLL is a digitally-controlled oscillator (DCO) whose frequency is tuned through the digital control of a bank of frequency-control

elements (FCEs). Each FCE is driven by a 1-bit digital sequence, and the DCO's frequency is adjusted by changing the state of one or more FCEs at a time [7], [8], [9].

The choice of the FCE's frequency step-size (i.e., the amount by which the DCO's frequency changes when the FCE's input bit changes) is subject to a trade-off between the DCO's quantization-error contribution to the PLL's phase noise and tuning range. To simultaneously achieve low phase noise and wide tuning ranges, DCOs with small frequency step-size and a large number of FCEs (256 to 1024 elements) are typically implemented [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

FCE banks comprised of power-of-two weighted elements require relatively small numbers of control signals, which is attractive as this reduces layout and routing complexity. Unfortunately, they suffer from non-monotonicities and propagation-delay timing mismatches between control signals, which introduce nonlinearity and frequency glitches [9], [13], [22]. In contrast, unit-weighted FCE banks require large numbers of control signals but are inherently monotonic and, in principle, immune to timing mismatches as the state of a single control signal changes at a time [9], [23]. Row-column control schemes reduce the number of signals needed to control unit-weighted banks [9], [11], [14], [15], [16]. However, to guarantee that only one of the row/column signals changes at a time, the number of control signals needed is still relatively large. For example, in [17], [18], and [19], 48 control lines were needed to control an 8-bit bank.

In this letter, an incremental frequency-control (IFC) scheme is presented for the control of unit-weighted FCE banks with arbitrary numbers of elements. The scheme uses a pair of differential 1-bit control signals where only one pair changes its state at a time, thus eliminating

timing-mismatch-related frequency glitches. Measurement results are presented to demonstrate the functionality of the proposed frequency control scheme and its negligible impact on a PLL's locking time and phase noise.

II. FREQUENCY CONTROL IN DCOs

In digital PLLs, an f_{ref}-rate digital sequence is used to control the DCO's instantaneous frequency by adjusting the states of the individual FCEs in the DCO's FCE bank, where f_{ref} is the frequency of the PLL's reference oscillator [9], [24]. In applications that require low phase noise, the required DCO frequency step-size, Δ, is in the order of tens of Hz, but practical existing FCEs have a typical minimum frequency step-size, Δ_{min}, of tens of kHz [7], [8], [10], [12], [21], [25].

Fig. 1.1 illustrates a common solution to this problem presented in the context of a 16-bit *LC*-oscillator [10]. Each FCE adds to or subtracts from the overall tank capacitance, and the minimum FCE frequency step-size is $\Delta_{min} = 2^8\Delta$. The f_{ref}-rate sequence $d[n]$ is split into two sequences, $d_I[n]$ and $d_F[n]$, comprised of the eight MSBs and eight LSBs of $d[n]$, respectively. The sequence $d_I[n]$ directly controls a bank of power-of-two-weighted FCEs labeled *integer FCE bank*. To achieve frequency step-sizes smaller than Δ_{min}, $d_F[n]$ is re-sampled to f_{fast} (with $f_{fast} > f_{ref}$), re-quantized, and the result is encoded to drive a unit-weighted FCE bank labeled *fractional FCE bank*. For instance, $d_F[n]$ can be re-quantized by a second-order digital delta-sigma modulator and the encoder can be a simple binary-to-thermometer encoder. Changes in $d_I[n]$ and $d_F[n]$ cause the DCO frequency to change by integer and fractional multiples of Δ_{min}, respectively, which is why the integer and fractional FCE banks are labeled as such.

For a given Δ_{min}, f_{fast} is chosen so that the digital re-quantization error does not degrade the PLL's phase noise. In addition, the integer FCE bank must be large enough to accommodate for the required DCO tuning range, typically in the range of tens of MHz. In [26], for instance, measured DCO frequency variations of about −200 kHz/°C are reported, so a 20 MHz tuning range would be needed to cover a 100 °C temperature range in such case. To meet both phase noise and tuning range requirements, FCE banks comprising 256 to 1024 unit elements are typically used [12], [14], [15], [16], [17], [18], [19], [20], [21].

The use of FCE banks comprised of power-of-two weighted FCEs allows for small numbers of control signals, which is convenient as this reduces routing complexity and the number of toggling digital lines routed near the DCO's tank. Ultimately, this reduces potential coupling of noise and spurious tones into the DCO output. Unfortunately, such FCE banks suffer from non-monotonicities and transient glitches that degrade a PLL's phase noise [9], [13], [21]. Non-monotonicities are more probable in power-of-two weighted banks with large number of elements and high mismatches between unit elements [9], [13], [23]. Unfortunately, this is the case with FCE banks as FCEs with small area and frequency step-size are needed to achieve fine frequency resolution and to minimize the phase noise degradation due to large routing-networks parasitcs. In addition, multiple control lines might need to toggle simultaneously, as in a 0…0011 to 0…0100 transition, and inevitable timing mismatches between the control lines result in transient glitches in the DCO output frequency.

Unit-weighted banks, in contrast, are inherently monotonic and immune to timing mismatches as only one control signal changes its state at a time [9], [23]. To mitigate the complexity associated with the large numbers of control signals needed by these banks, row-

column control schemes have been proposed. Such schemes have enabled, for example, the control of an 8-bit bank with as little as 16 digital control signals [10], [11], [12], [14]. The conventional row-column control approach, however, suffers from the timing mismatch problem because the change of state of corner units requires more than one control signal changing simultaneously [9], [17]. In [17], [18], and [19], different control signals are used for cells in even and odd columns to guarantee that a single control signal changes its state at a time. Unfortunately, this solution requires additional redundant control signals, increasing the number of control lines for an 8-bit bank, for example, to 48 lines.

III. PROPOSED IFC SCHEME

The proposed IFC scheme, shown in Fig. 1.2(a) in the context of an *LC*-DCO with an 8-bit unit-weighted integer FCE bank, both minimizes the number of control signals and ensures that the state of at most one control signal changes at a time [27]. The integer and fractional FCE banks in Fig. 1.2(a) are connected to the core of the *LC* oscillator as in Fig. 1.1, and the fractional FCE bank is controlled by $d_F[n]$ as described in Section II. The integer FCE bank, comprised of unit-weighted FCEs, is controlled by a pair of 1-bit control signals, $c_1[r]$ and $c_2[r]$, where only one of them changes its state at a time. This reduces the control and routing complexity of the bank, and eliminates timing mismatch related glitches.

A. *Qualitative Description*

The proposed IFC scheme operates as follows. In unit-weighted banks, the number of FCEs with control bit of one must be equal to $d_I[n]$. Instead of directly driving the integer FCE

bank with $d_I[n]$, this sequence is first re-sampled to f_{fast}, producing $d_I[r]$. At each rising edge of the fast clock, $d_I[r]$ is compared to the current number of FCEs being driven with a one, $t[r-1]$, and the difference is used to compute the number of one-step increments or decrements that need to be issued so that $t[r] = d_I[r]$. The integer FCE bank is then updated by changing the state of at most one FCE per fast clock period, so only one f_{fast}-rate 1-bit control signal changes its state at a time.

Fig. 1.2(b) shows the integer FCE bank's top-level structure. As explained below, the layer of switches acts as an interface between $c_1[r]$ and $c_2[r]$ and the FCEs, and guarantees that at any given moment $c_1[r]$ and $c_2[r]$ are each connected to a single FCE. A change in the state of one of the control signals increments $t[r]$ by one, and a change in the state of the other control signal decrements $t[r]$ by one. The control signals $c_1[r]$ and $c_2[r]$ are generated within the DCO's digital interface which comprises an incremental-switching logic (ISL) block and a finite-state machine (FSM). The ISL generates the sequence $d_I[r]$, compares it to $t[r-1]$, and outputs the sequence $m[r]$ that takes on a value of -1, 0, or 1, commanding a decrease in the DCO's frequency by Δ_{min} (*dn*), no frequency change (*noc*), or an increase by Δ_{min} (*up*), respectively. The FSM generates $c_1[r]$ and $c_2[r]$, each taking on a value of 0 or 1, based on $m[r]$, $c_1[r-1]$ and $c_2[r-1]$.

Fig. 1.2(c) shows $d_I[n_t]$, $m[r_t]$, and how the FCEs' states change accordingly, where $n_t = n$ over the nth period of the f_{ref}-rate clock that updates $d_I[n]$ and $r_t = r$ over the rth period of the f_{fast}-rate clock that updates $m[r]$.[1] In this example, $f_{fast} = 3f_{ref}$ and shaded boxes are used to

[1] By definition, n_t is the largest integer less than or equal to $f_{ref}t$ at time t, so it is a continuous-time waveform. Hence, $d_I[n_t]$ is a continuous-time waveform even though $d_I[n]$ is a discrete-time sequence. The same applies for $m[r_t]$ and $m[r]$ with n replaced by r, and f_{ref} replaced by f_{fast}.

indicate FCEs with a control bit of one. At $t = kT_{ref}$, $d_I[n_t]$ changes from 0 to 1 and so does $m[r_t]$, dictating FCE 1 to increase the DCO's frequency by Δ_{min}. At $t = kT_{ref} + T_{fast}$, $m[r_t]$ goes low, signaling that no more FCE states need to be changed. The opposite happens between $(k+1)T_{ref}$ and $(k+2)T_{ref}$ where the state of FCE 1 is changed such the DCO's frequency decreases by Δ_{min}. At $t = (k+2)T_{ref}$, $d_I[n_t]$ changes from 0 to 2. As the state of a single FCE changes at a time, $m[r_t]$ stays high over two fast clock periods where the states of FCEs 2 and 3 are changed sequentially at $t = (k+2)T_{ref}$ and $t = (k+2)T_{ref} + T_{fast}$, respectively.

B. *Architecture Details*

Fig. 1.3 shows the details of the ISL block. A flip-flop resamples $d_I[n]$ at f_{fast} to generate $d_I[r]$. As $m[r] = 1$ indicates a frequency increment, $t[r-1]$, is equal to the running sum of $m[k]$ from $k = 0$ through $k = r - 1$. The result of $d_I[r] - t[r-1]$ is clipped to ±1 and a non-zero carry signal is generated if the required number of increments/decrements exceeds one, and adds to $d_I[r]$ in the next cycle. It is worth noting that in high-performance PLLs, the frequency noise is significantly lower than Δ_{min} after locking, so $d_I[n]$ should not change by more than ±1 over a reference period in such case. Hence, the serialization of increments/decrements associated with the proposed scheme does not affect the PLL's operation or phase noise after locking. Moreover, as shown in Section V, the proposed scheme also has a negligible impact on a PLL's locking behavior.

Fig. 1.4(a) shows the top-level diagram of an integer FCE bank architecture that is compatible with the IFC scheme, where LFCE stands for latched-FCE. An LFCE is a regular FCE whose control voltage, $v_i[r]$, is latched by a pair of cross-coupled tri-state inverters. Fig. 1.4(b) shows the topology of two types of LFCEs, LFCE-0 and LFCE-1, that are identical

except that in LFCE-0/1, $v_i[r] = 0/1$ results in an increase in the DCO's frequency. Additionally, each LFCE has a pair of tri-state inverters that are enabled by a global reset signal (*rst*) and are used to initialize the bank to a hard-coded initial condition (*ic*). Each LFCE is connected to either $c_1[r]$ or $c_2[r]$ in an alternating fashion through two switches in series, with the top switch controlled by the state of the LFCE to the right and the bottom switch controlled by the state of the LFCE to the left. At any given moment, only two LFCEs, referred to as "Active LFCEs", are accessed by $c_1[r]$ or $c_2[r]$, allowing for the state of a single FCE to change at a time. The LFCE topology makes sure that each FCE is properly initialized and preserves its control logic value when disconnected from its respective control signal.

To ensure that only one of $c_1[r]$ and $c_2[r]$ change its state at a time, the LFCE types are arranged in a …, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, … pattern, and the highlighted FCEs' states in Fig. 1.4(a) correspond to a scenario where the four FCEs to the left have a control bit of 1 and the four FCEs to the right have a control bit of 0. Consequently, the configuration of the series switches between each LFCE and either of the control signals follows a pattern that repeats every four branches. This pattern choice guarantees that at any given moment only two LFCEs are accessed by $c_1[r]$ or $c_2[r]$, and for any transitioning LFCE, the LFCEs to its right and left have the same latched voltage. Hence, if $c_1[r]$ were to change, $c_2[r]$ does not have to change as the LFCE that is currently connected to $c_2[r]$ and the LFCE that will be connected to $c_2[r]$ when $c_1[r]$ changes have the same latched voltage.

Fig. 1.5 shows the state-transition diagram of the FSM. It has four states corresponding to the possible combinations of c_1 and c_2 and generates $c_1[r]$ and $c_2[r]$, each taking on a value of 0 or 1, based on $m[r]$, $c_1[r-1]$ and $c_2[r-1]$. As only one of the control signals changes at a

time, the two-bit binary word formed by concatenating $c_1[r]$ and $c_2[r]$ follows a Gray-encoder pattern. This is a result of the LFCEs pattern choice.

As the integer FCE bank architecture employs unit-weighted FCEs where the state of a single control signal and FCE are changed at a time, the IFC scheme is inherently monotonic and free of frequency glitches across the whole bank.

IV. IMPLEMENTATOIN DETAILS

The IFC scheme was implemented as a modification to the digital fractional-*N* PLL presented in [20]. As the details of the PLL are explained in [20], only the additional implementation details relevant to the IFC scheme are presented here. The IC was implemented in the Global-Foundries 22-nm FDSOI process and consists of a digital fractional-*N* PLL, a serial peripheral interface (SPI), and additional circuitry used for testing. The PLL comprises five main blocks: crystal oscillator, core analog circuitry, place-and-route (PNR) digital block, DCO, and output drivers. All blocks run from a 0.8-V power supply, except for the output drivers that use a 1-V power supply.

A. *DCO Digital Interface*

The DCO's digital interface has the form in Fig. 1.2(a), with the ISL block signal processing details shown in Fig. 1.3 and the FSM implementing the state-transition diagram in Fig. 1.5. The FSM outputs $c_1[r]$ and $c_2[r]$ and their inverted versions, enabling differential control for the integer FCE bank's unit cells. The digital re-quantizer in Fig. 1.2(a) is implemented as a successive re-quantizer with eight quantization blocks and first-order high-

pass shaped quantization error [28]. It generates an f_{fast}-rate sequence, $c_F[r]$, that takes on values of 0 or 1 and controls the state of a single FCE that plays the role of the fractional FCE bank (hence, no encoder is implemented). As in [20] and [26], an integer-boundary avoider can be incorporated at the interface between $d[n]$ and $d_I[n]$ and $d_F[n]$ without affecting the proposed IFC scheme. The DCO digital interface circuitry is implemented as a part of the PLL's PNR digital block that is clocked at f_{fast}. The FSM and digital re-quantizer outputs are resynchronized to the fast clock within the PNR digital block before being routed to the DCO.

B. *FCE Banks*

The DCO core circuitry is the same as in [20] except for the FCE banks. The DCO has a single FCE playing the role of the fractional FCE bank and the integer FCE bank has the architecture in Fig. 1.4(a) and comprises 256 unit elements.

The integer FCE bank in Fig. 1.4(a) comprises blocks of four different unit cells that repeat across the bank. Fig. 1.6(a) highlights the four unit cells A1, A0, B0, B1, each controlled differentially by $c_1[r]$, $\overline{c}_1[r]$, $c_2[r]$, and $\overline{c}_2[r]$. The bank is laid out as illustrated in Fig. 1.6(b). The elements are placed in eight rows, each containing eight {A1, A0, B0, B1} blocks, and are surrounded by dummy FCEs to improve matching. The rows are connected in a zigzag manner as indicated by the dotted arrows between the B1 and A1 units at the bank's edges.

The control signals $c_1[r]$, $\overline{c}_1[r]$, $c_2[r]$, and $\overline{c}_2[r]$ are routed close to each other from the digital PNR block to the bottom-middle part of the bank, and then in a tree-structured manner to both sides of the bank where they get resynchronized to the fast clock and buffered. The resynchronization aims to switch the state of the FCEs as close as possible to the DCO's differential zero-crossings to minimize disturbances to the tank's energy [10]. Routing the

control signals from both sides of the bank minimizes the delay from the control signals to the individual cells. Although not shown in Fig. 1.6(b), one of the dummy FCEs is used as the fractional FCE bank and is controlled by $c_F[r]$ that is also locally resynchronized to the fast clock. The FCE banks occupy an area of 120×42 μm^2, compared to 120×30 μm^2 in [20]. The banks are surrounded by a 10-μm wide isolation region with reduced substrate doping.

C. *Integer FCE Bank Unit Cell*

Figs. 1.6(c)-(f) show the circuit implementation details of cells A1, A0, B0, and B1, respectively. Each cell comprises an FCE unit, four tri-state inverters, and four switches. The four cells have the same topology except for the signals that control the switches and the FCE.

Initial conditions are hard-wired within each cell such that on reset one half of the FCE units add to the DCO capacitance. During initialization, c_1 and c_2 are set to 0 and 1, respectively, and one of the tri-state inverters in the latch is disabled to avoid fighting between latches in different cells that might occur as the switches settle to their desired configuration.

The tri-state inverters and switches sizing takes advantage of the FDSOI IC technology where the back gates of all PMOS transistors are tied to ground to reduce threshold voltages and increase speed. This allowed for the design of switches with equally-sized PMOS and NMOS transistors (aspect ratio of 420nm/20nm) and the use of minimum-sized standard-cell tri-state inverters, reducing the unit cell's area. The differential control of individual units reduces the coupling between the control signals and nearby interconnects. Also, it reduces the fighting between the latches within each unit cell and the control signal drivers at both sides of the bank, relaxing the drivers' speed and power requirements.

Fig. 1.7(a) and (b) show the FCE topology and its switch implementation, respectively [8]. Changing the gate voltage of M_{sw} changes the capacitance seen across nodes v_{DCO+} and v_{DCO-}, and the other two transistors are biased in the triode region to set the DC value of nodes v_t and v_b. Each FCE creates a simulated capacitance step of 63 aF, equivalent to Δ_{min} = 137 kHz at 6.7 GHz. The small incremental capacitance provided by the FCE structure results in a quality-factor over 200, allowing for the size of M_{sw} to be only 2× the minimum transistor size. The 256 units cover a simulated tuning range of 35 MHz.

Fig. 1.7(c) shows the layout of a unit cell in the integer FCE bank. The four capacitors are implemented as custom metal-oxide-metal (MOM) structures, and the circuitry is placed underneath them. Each of the capacitors is formed by sandwiching eleven metal 5 fingers between two metal planes drawn in metals 4 and 6. The blue drawing in Fig. 1.7(c) represents the metal 5 fingers, and the four yellow solid rectangles represent the metal 4 planes underneath the metal 5 fingers. Four metal 6 solid planes (not shown) atop the metal 5 fingers and aligned with the four metal 4 planes complete the structure for each of the capacitors. Rectangular stripes in metal 7 are used to tap the FCE terminals to the DCO outputs. As the FCE quality factor is high, metals 4 through 6 were chosen to build the capacitor, sparing the upper low-resistance metals for top-level routing to avoid degrading the DCO's tank quality factor. Metal 3 is used to form a ground shield around each unit, which minimizes the interaction between adjacent FCEs. Metals 1 and 2 are used for routing between all other circuitry underneath the FCE structure. Each unit cell occupies an area of 2.9×3.6 μm^2.

V. MEASUREMENT RESULTS

A die photograph of the prototype is shown in Fig. 1.8. The reference frequency, f_{ref}, is 80 MHz and $f_{fast} = f_{PLL}/8 \approx 835$ MHz, which is synchronous to the DCO output. The measured power consumption of the integer FCE bank's digital circuitry is 67 μW.

Fig. 1.9 shows a representative measured PLL phase noise profile at f_{PLL} = 6.67 GHz for a PLL bandwidth of 550 kHz. The jitter and phase noise profile are in line with those reported in [20], indicating that the proposed IFC scheme has minimal impact on the PLL's phase noise. At 6.67 GHz, the measured DCO fine tuning range is 37.6 MHz corresponding to Δ_{min} = 147 kHz.

To evaluate the impact of the proposed IFC scheme on the PLL's locking behavior, the DCO's free-running frequency was set manually to 6.69 GHz and the PLL configuration was set such that when locked, the DCO frequency settles around 6.67 GHz. The DCO control was then switched back from manual control to the PLL, and the DCO's output frequency was recorded. The same experiment was replicated using behavioral simulations for two cases, one with ideal DCO control, i.e., the DCO frequency is $d[n]$ scaled by $2^{-8}\Delta_{min}$, and another with the IFC scheme. Fig. 1.10(a) shows the measured frequency settling behavior of the PLL and Fig. 1.10(b) shows the simulation results. These results show that the PLL's measured locking behavior is close to the predicted locking behavior from simulations, and they show that the locking behavior using the proposed IFC scheme is nearly identical to that with an ideal frequency control scheme.

Additional behavioral simulations were used to evaluate the impact of the IFC scheme on the PLL locking for different PLL initial conditions. Fig. 1.11(a) and (b) show histograms

of the PLL locking time using the proposed IFC scheme and an ideal control scheme, respectively, for 10,000 PLL runs. For each run, the PLL was initialized to have a random crystal oscillator initial phase and an initial DCO frequency error between −10 MHz and 10 MHz. For the results in Fig. 1.11(a) and (b), locking time is defined as the time after which $d_{\mathrm{I}}[n]$ does not change by more than ±1 for at least 2000 consecutive reference cycles. The results show the same average locking time and statistics for both schemes. The difference in the locking times between the IFC scheme and the ideal DCO control scheme was also measured for each of the 10,000 PLL runs. Fig. 1.11(c) shows a histogram of the results, indicating an average locking time difference of 1.7 ns with a standard deviation of 0.2 μs, thus verifying the negligible impact the IFC scheme has on the PLL's locking behavior.

VI. CONCLUSION

In this letter, an IFC scheme for DCOs tuning is presented. The scheme uses a pair of differential 1-bit control signals to control an arbitrarily large bank of unit-weighted FCEs, where at most one of the control signals changes its state a time. This guarantees monotonicity and eliminates frequency glitches typically caused by inevitable timing mismatches between multiple control signals. Measurement results are presented to validate the functionality of the proposed IFC scheme and verify the negligible impact it has on a PLL's locking behavior and phase noise.

ACKNOWLEDGEMENTS

This chapter, in part, is currently being prepared for submission for publication of the material. A. I. Eissa, E. Alvarez-Fontecilla, E. Helal, I. Galton. The dissertation author is the primary investigator and author of this material. Professor Ian Galton supervised the research which forms the basis for this material.

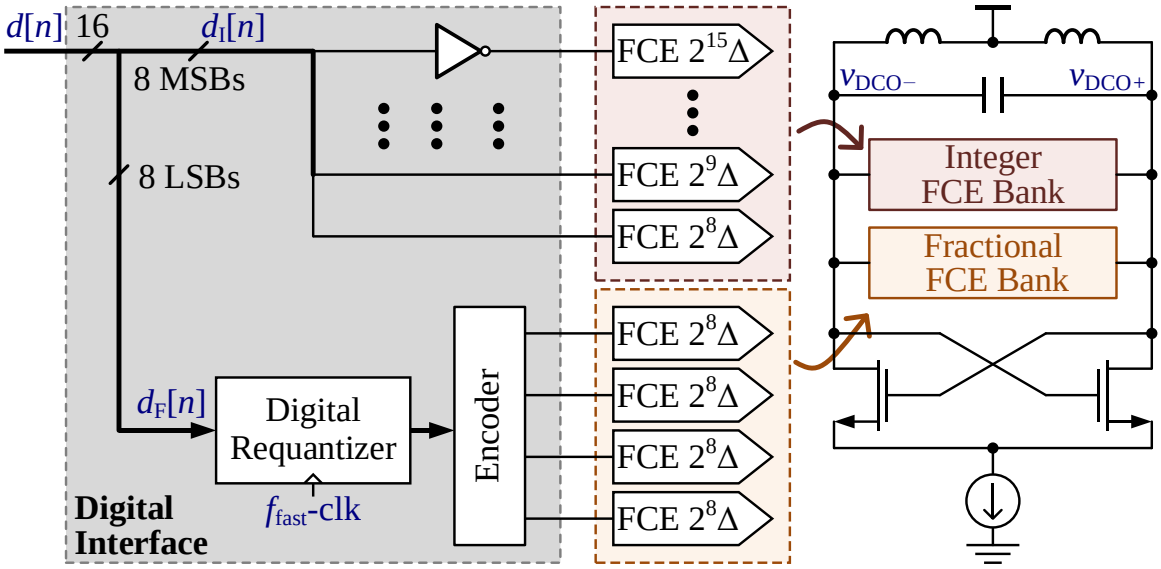


Figure 1.1. Conventional DCO frequency control technique.

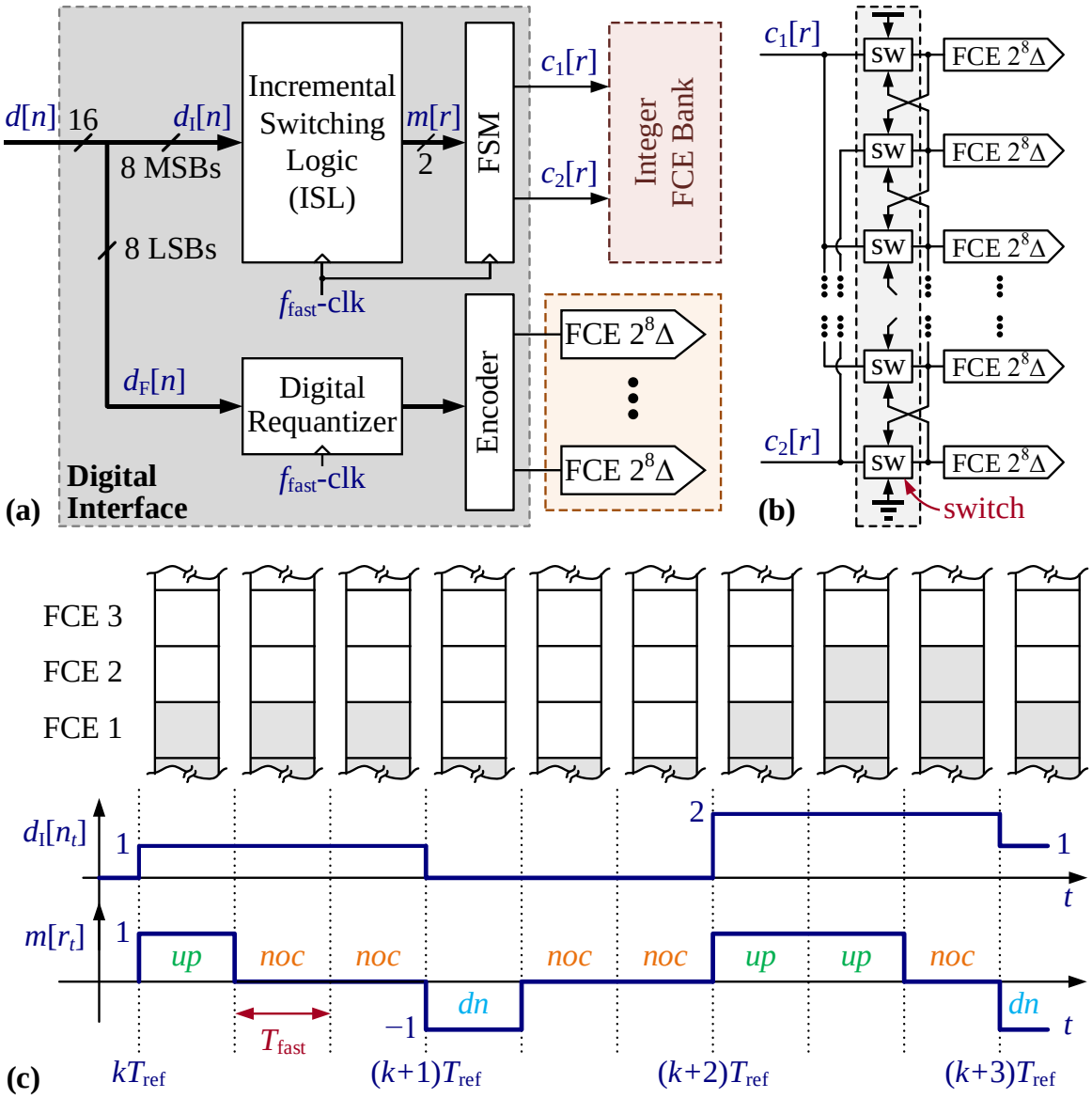


Figure 1.2. (a) Top-level block diagram of the proposed IFC scheme, (b) integer FCE bank top-level structure, and (c) example waveforms for $d_I[n_t]$ and the FSM output, $m[r_t]$, along with an illustration of the FCEs' states.

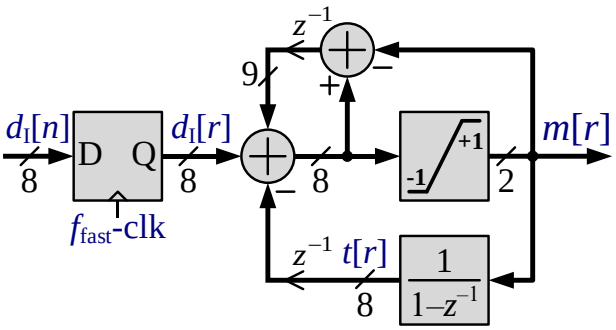


Figure 1.3. Incremental switching logic signal processing details.

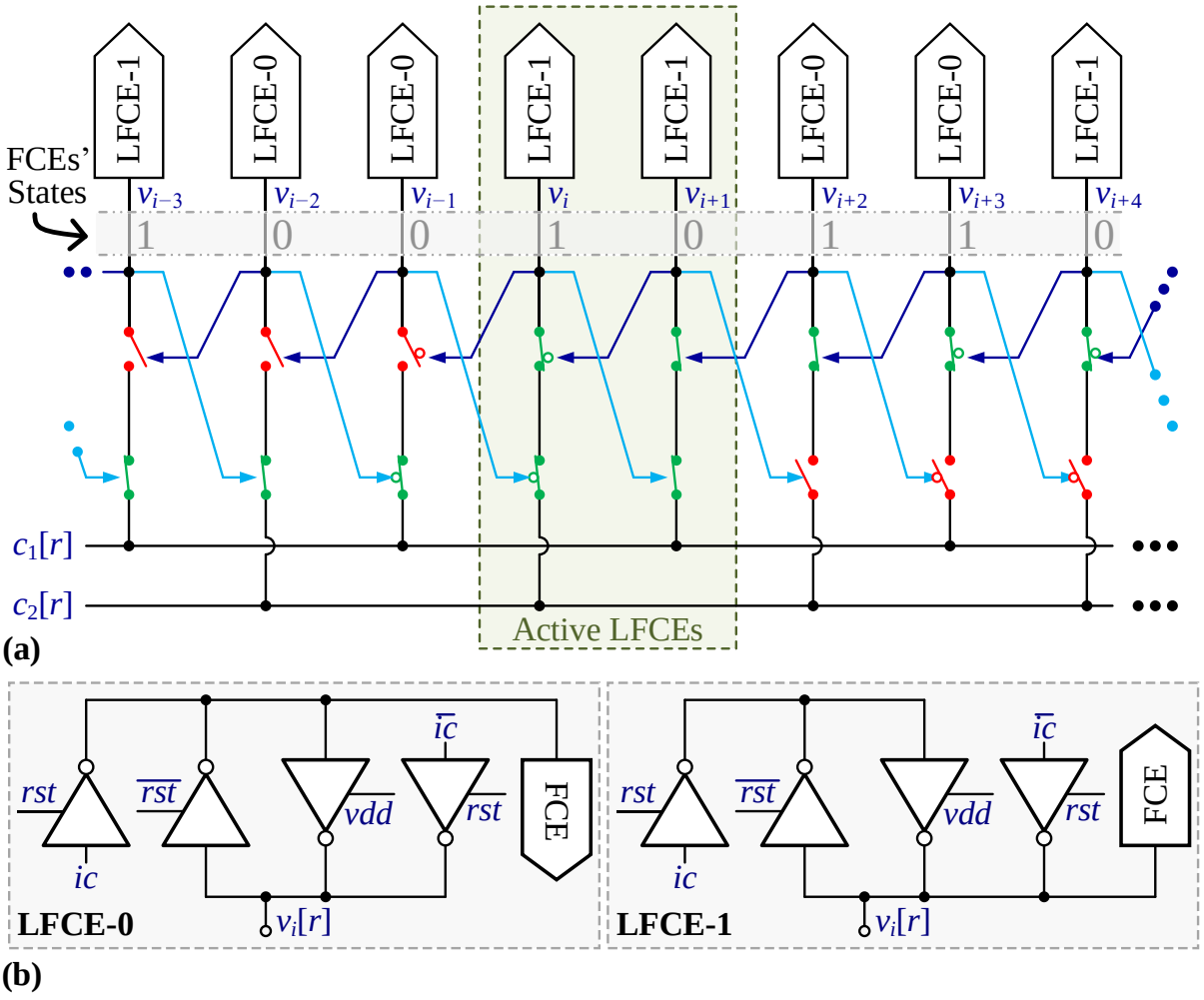


Figure 1.4. Proposed integer FCE bank architecture compatible with the IFC scheme.

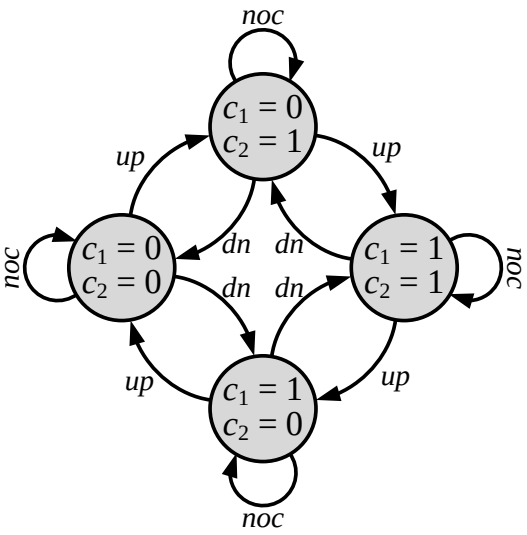


Figure 1.5. Incremental switching FSM state-transition diagram.

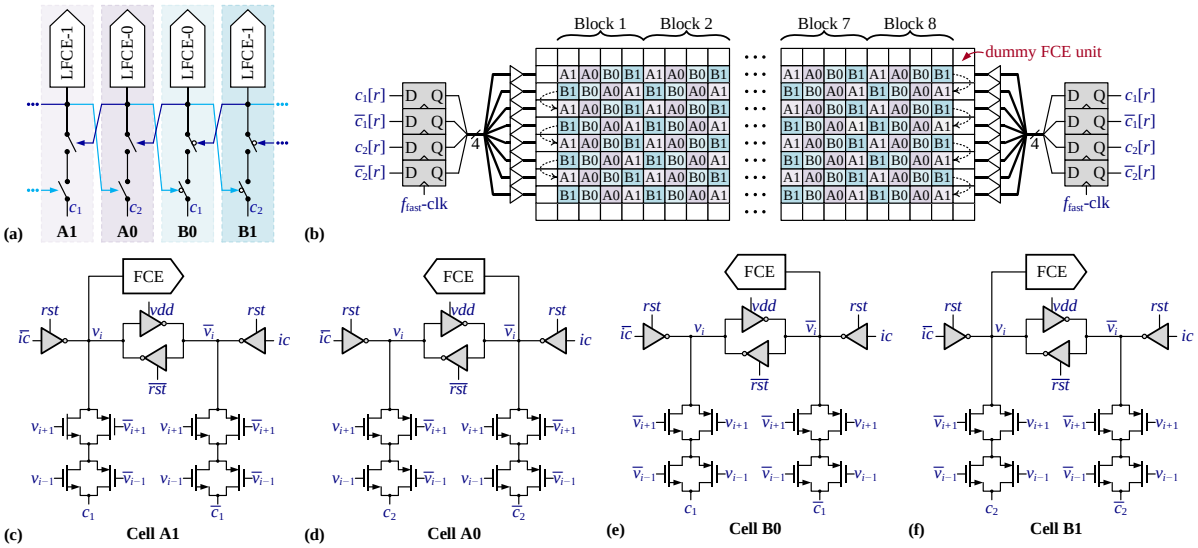


Figure 1.6. (a) The four main unit cells comprising the integer FCE bank, (b) integer FCE bank layout, and (c) – (f) circuit implementation details of unit cells A1, A0, B0, and B1, respectively.

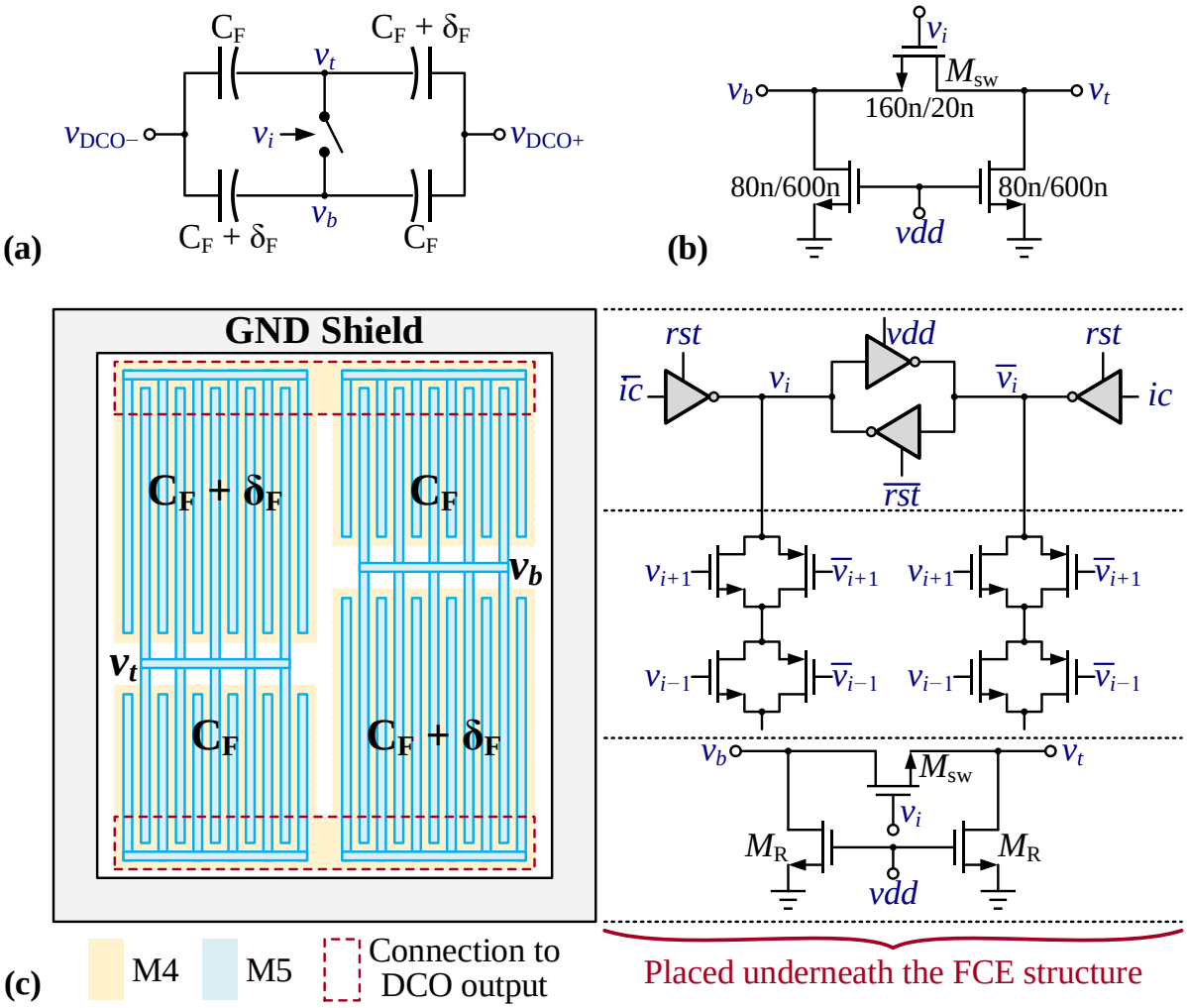


Figure 1.7. (a) FCE circuit topology, (b) FCE switch implementation, and (c) integer FCE bank unit cell layout.

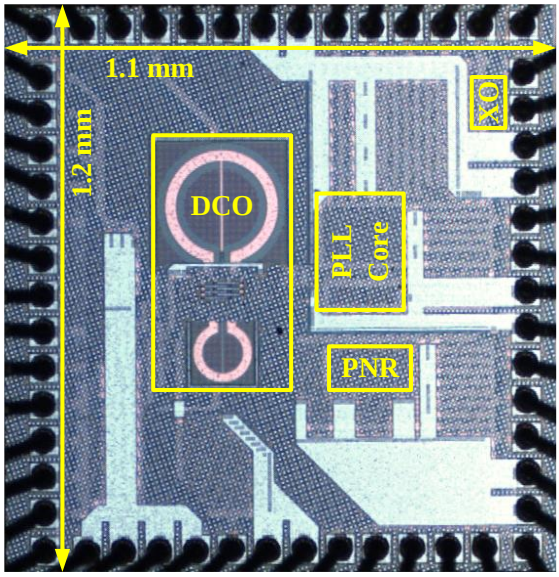


Figure 1.8. Die photograph.

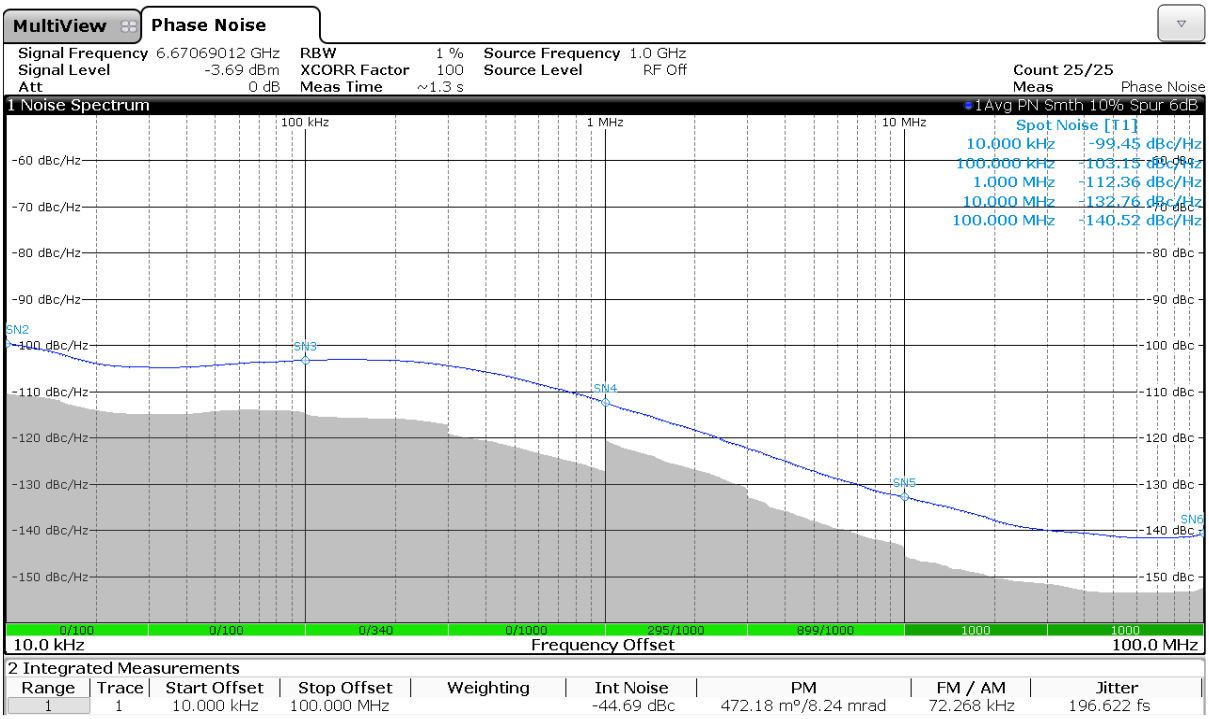


Figure 1.9. Measured PLL phase noise at f_{PLL} = 6.67 GHz for a 550-kHz loop bandwidth.

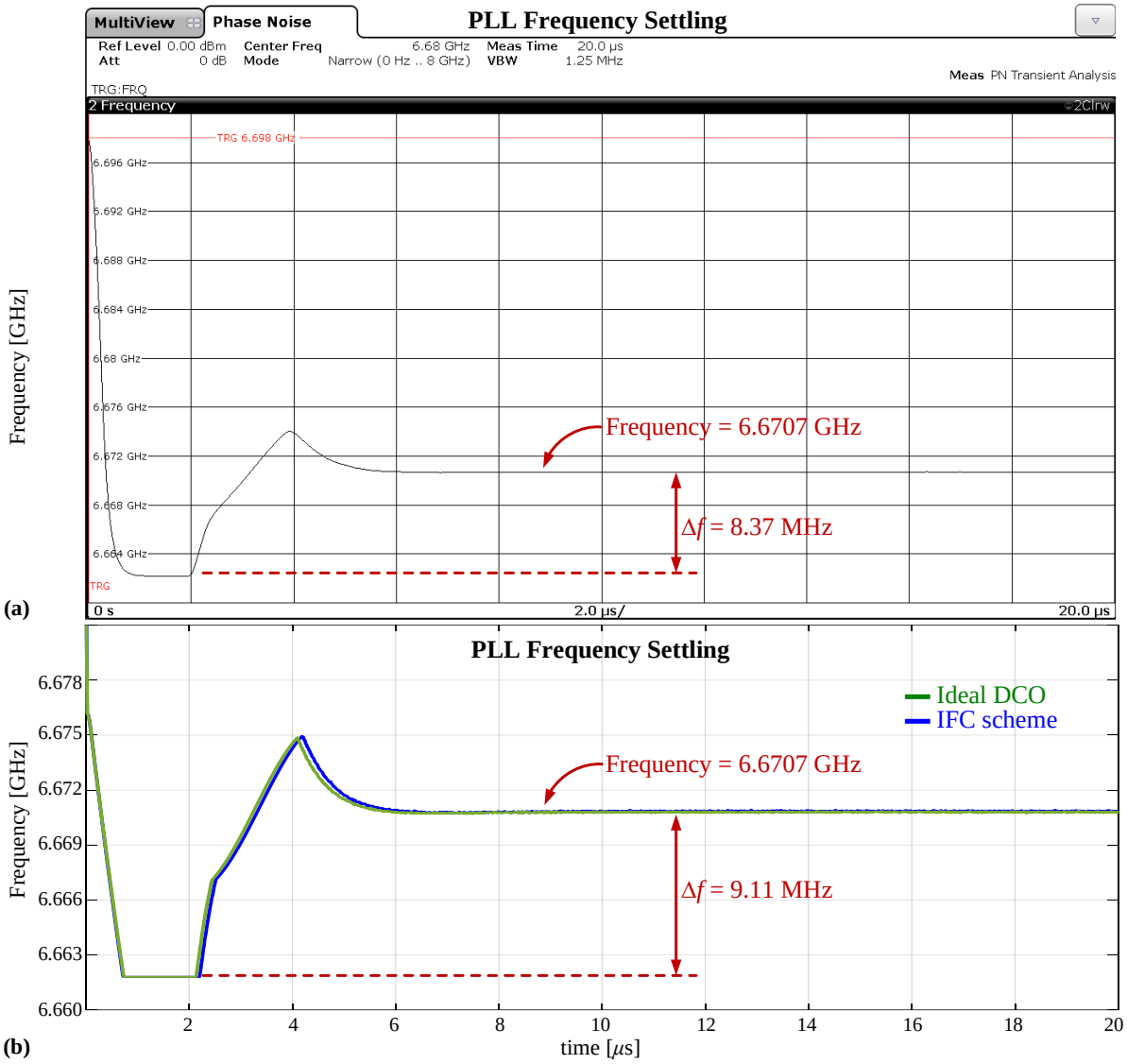


Figure 1.10. PLL frequency-settling vs time. (a) Measurement results, and (b) behavioral model simulation results.

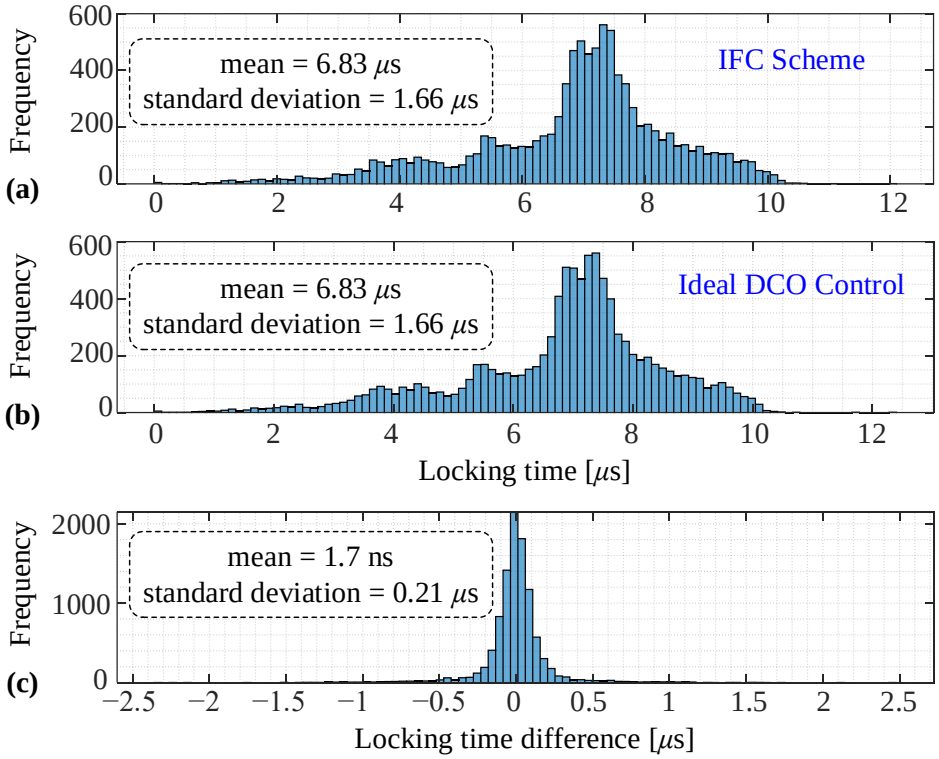


Figure 1.11. (a), (b) PLL locking time histogram using the proposed IFC scheme and ideal DCO control, respectively, and (c) locking time difference histogram.

REFERENCES

1. C. -R. Ho and M. S. -W. Chen, "Smoothing the Way for Digital Phase-Locked Loops: Clock Generation in the Future with Digital Signal Processing for Mitigating Spur and Interference," in *IEEE Microwave Magazine*, vol. 20, no. 5, pp. 80-97, May 2019.

2. Y. -L. Hsueh *et al*., "28.2 A 0.29mm2 frequency synthesizer in 40nm CMOS with 0.19psrms jitter and <-100dBc reference spur for 802.11ac," *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, San Francisco, CA, USA, 2014, pp. 472-473.

3. C. -W. Yao *et al*., "A 14-nm 0.14-psrms Fractional-N Digital PLL With a 0.2-ps Resolution ADC-Assisted Coarse/Fine-Conversion Chopping TDC and TDC Nonlinearity Calibration," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 12, pp. 3446-3457, Dec. 2017.

4. S. Levantino, "Recent Advances in High-Performance Frequency Synthesizer Design," *2022 IEEE Custom Integrated Circuits Conference (CICC)*, Newport Beach, CA, USA, 2022, pp. 1-7.

5. F. Buccoleri *et al*., "A 72-fs-Total-Integrated-Jitter Two-Core Fractional-N Digital PLL With Digital Period Averaging Calibration on Frequency Quadrupler and True-in-Phase Combiner," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 3, pp. 634-646, March 2023.

6. S. M. Dartizio *et al*., "A Low-Spur and Low-Jitter Fractional-N Digital PLL Based on an Inverse-Constant-Slope DTC and FCW Subtractive Dithering," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 12, pp. 3320-3337, Dec. 2023.

7. R. B. Staszewski, Chih-Ming Hung, D. Leipold and P. T. Balsara, "A first multigigahertz digitally controlled oscillator for wireless applications," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 11, pp. 2154-2164, Nov. 2003.

8. C. Venerus and I. Galton, "A TDC-Free Mostly-Digital FDC-PLL Frequency Synthesizer With a 2.8-3.5 GHz DCO," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 2, pp. 450-463, Feb. 2015.

9. B. Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge: Cambridge University Press, 2020.

10. R. B. Staszewski, D. Leipold, K. Muhammad and P. T. Balsara, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS Process," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 815-828, Nov. 2003.

11. R. B. Staszewski, Chih-Ming Hung, N. Barton, Meng-Chang Lee and D. Leipold, "A digitally controlled oscillator in a 90 nm digital CMOS process for mobile phones," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2203-2211, Nov. 2005.

12. L. Fanori, A. Liscidini and R. Castello, "3.3GHz DCO with a frequency resolution of 150Hz for All-digital PLL," *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2010.
13. I. Bashir, R. B. Staszewski and P. T. Balsara, "A Digitally Controlled Injection-Locked Oscillator With Fine Frequency Resolution," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 6, pp. 1347-1360, June 2016.

14. T. Pittorino, Y. Chen, V. Neubauer, T. Mayer and L. Maurer, "A UMTS-complaint fully digitally controlled oscillator with 100Mhz fine-tuning range in 0.13/spl mu/m CMOS," *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, San Francisco, CA, USA, 2006.

15. J. Lin *et al*., "A PVT tolerant 0.18MHz to 600MHz self-calibrated digital PLL in 90nm CMOS process," *2004 IEEE International Solid-State Circuits Conference (IEEE Cat. No.04CH37519)*, San Francisco, CA, 2004.

16. N. Da Dalt, E. Thaller, P. Gregorius and L. Gazsi, "A compact triple-band low-jitter digital LC PLL with programmable coil in 130-nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 7, pp. 1482-1490, July 2005.

17. N. Da Dalt, C. Knopf, M. Burian, T. Hartig and Hermann Eul, "A 10b 10GHz digitlly controlled LC oscillator in 65nm CMOS," *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, San Francisco, CA, USA, 2006.

18. A. Elkholy, M. Talegaonkar, T. Anand and P. Kumar Hanumolu, "Design and Analysis of Low-Power High-Frequency Robust Sub-Harmonic Injection-Locked Clock Multipliers," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 12, pp. 3160-3174, Dec. 2015.

19. A. Elkholy, A. Elmallah, M. G. Ahmed and P. K. Hanumolu, "A 6.75–8.25-GHz −250-dB FoM Rapid ON/OFF Fractional-N Injection-Locked Clock Multiplier," in *IEEE Journal of Solid-State Circuits*, vol. 53, no. 6, pp. 1818-1829, June 2018.

20. E. Helal, E. Alvarez-Fontecilla, A. I. Eissa and I. Galton, "A Time Amplifier Assisted Frequency-to-Digital Converter Based Digital Fractional-N PLL," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2711-2723, Sept. 2021.

21. E. Alvarez-Fontecilla, E. Helal, A. I. Eissa and I. Galton, "Spectral Breathing and Its Mitigation in Digital Fractional-N PLLs," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 10, pp. 3191-3201, Oct. 2021.

22. S. Akhtar, M. Ipek, J. Lin, R. B. Staszewski and P. Litmanen, "Quad Band Digitally Controlled Oscillator for WCDMA Transmitter in 90nm CMOS," *IEEE Custom Integrated Circuits Conference 2006*, San Jose, CA, USA, 2006.

23. Chi-Hung Lin and K. Bult, "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm^2," in *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1948-1958, Dec. 1998.
24. R. B. Staszewski and P. T. Balsara, "Phase-domain all-digital phase-locked loop," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 159-163, March 2005.

25. J. Zhuang, K. Waheed, and R.-B. Staszewski, “Design of spur-free frequency tuning interface for digitally controlled oscillators,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 62, no. 1, pp. 46–50, Jan. 2015.

26. C. Weltin-Wu, G. Zhao and I. Galton, "A 3.5 GHz Digital Fractional- PLL Frequency Synthesizer Based on Ring Oscillator Frequency-to-Digital Conversion," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 12, pp. 2988-3002, Dec. 2015.

27. I. Galton, E. Alvarez-Fontecilla, and A. I. Eissa, “Frequency to digital converter, asynchronous phase sampler and digitally controlled oscillator methods,” U.S. Patent 11437980, Sep. 6, 2022.

28. A. Swaminathan, A. Panigada, E. Masry and I. Galton, "A Digital Requantizer With Shaped Requantization Noise That Remains Well Behaved After Nonlinear Distortion," in *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5382-5394, Nov. 2007.

CHAPTER 2

A DUTY-CYCLE-ERROR-IMMUNE REFERENCE FREQUENCY DOUBLING TECHNIQUE FOR FRACTIONAL-*N* DIGITAL PLLs

Abstract— Increasing a PLL's reference frequency offers significant performance advantages, but doing so by increasing the PLL's crystal oscillator frequency is not a viable option in many applications. Instead, a frequency doubler can be used to derive a reference signal with twice the frequency of the crystal oscillator, but conventional PLLs are highly sensitive to the crystal oscillator's duty cycle error in such cases. Prior solutions to this problem involve calibration techniques which impose convergence speed versus accuracy tradeoffs. In contrast, this paper proposes a system modification which makes a PLL immune to such duty cycle errors without the need for calibration. The technique is presented and analyzed in the context of a delta-sigma frequency-to-digital converter (ΔΣ-FDC) based PLL. Analysis and behavioral simulations with nonideal circuit parameters show that the worst-case convergence time is at least 10 times faster than that of the prior techniques. Additionally, the proposed ΔΣ-

Manuscript received 19 March 2024. This work was supported by Qualcomm, Analog Devices, and by the National Science Foundation under Award 1617545.

A. I. Eissa and I. Galton are with the Electrical and Computer Engineering Department, University of California, San Diego, La Jolla, CA 92092, USA (e-mail: aeissa@ucsd.edu). A. I. Eissa is also with the Electronics and Communications Engineering Department, Ain Shams University, Cairo, Egypt.

E. Alvarez-Fontecilla is with Analog Devices Inc., San Diego, CA 92128 USA and C. Weltin-Wu is with Aeonsemi, Inc., Santa Clara, CA 95134 USA.

FDC includes other modifications which improve its performance relative to comparable prior ΔΣ-FDCs.

I. INTRODUCTION

Phase-locked loops (PLLs) are critical components in communication systems, and their performance requirements continue to increase as communication system standards evolve. In particular, the demand for PLLs with sub-100-fs rms jitter is increasing to enable higher data rates in wireless and wireline communication systems [1], [2], [3], [4], [5], [6]. Furthermore, reciprocal-mixing requirements in some wireless applications require PLLs with reference spurs below –80 dBc.

A PLL's phase noise spectrum usually is dominated by the phase noise of its controlled oscillator above the PLL's bandwidth and by noise from all other circuitry within the PLL's bandwidth.[3] The PLL's in-band phase noise consists of white and highpass shaped components, that are essentially sampled at the reference frequency, f_{ref}, so doubling f_{ref} for a given PLL bandwidth reduces the contribution to the PLL's phase noise of the white and first-order highpass shaped noise components by 3 dB and 9 dB, respectively [7]. This reduces the PLL's jitter by reducing the in-band noise, and makes it possible to further reduce the jitter by increasing the PLL's bandwidth to suppress the controlled oscillator's phase noise contribution over the wider bandwidth.

[3] The controlled oscillator is a digitally-controlled oscillator in the case of a digital PLL and a voltage-controlled oscillator in the case of an analog PLL.

However, a PLL is generally but one component of a larger system, and its reference frequency is typically derived from the system's crystal oscillator. Unfortunately, the crystal oscillator frequency, $f_{crystal}$, is usually dictated by cost and system-level constraints, so increasing $f_{crystal}$ is rarely an option when designing the PLL. Instead, a frequency doubler (FD), which uses the rising and falling edges of the crystal oscillator to generate a double-frequency reference signal, can be used to effectively double f_{ref}. The drawback of the approach is that crystal oscillators typically have duty cycle errors of 5 to 10% across process, voltage, and temperature (PVT) variations, and conventional PLLs with FDs are highly sensitive to such errors [8], [9], [10], [11], [12]. The duty-cycle error results in large spurs at integer-multiples of $f_{crystal}$ and increase the PLL's jitter.

For example, a PLL with closed-loop bandwidth of 1.5 MHz with a 20dB/decade roll-off up to 76.8 MHz, a 10 GHz output frequency, and a 153.6 MHz reference signal from a 76.8 MHz crystal-oscillator with 5% duty-cycle error followed by an FD would have a −21.8 dBc spur at 76.8 MHz. This corresponds to 1.82 ps of jitter, not including any other error sources. Furthermore, the duty cycle error increases the dynamic range requirements of several of the PLL's circuit blocks, which generally increases their contributions to the PLL's phase noise and spurs.

Previously published techniques that address this problem rely on estimating the duty-cycle error in the analog or digital domains and canceling it through the crystal oscillator and FD analog circuitry, as in [13], [14], [15], [16], [17], and [18], or through the PLL's multi-modulus divider (MMD) as in [11], [12], [19], and [20]. However, these techniques are subject to a fundamental trade-off between convergence speed and accuracy. To sufficiently reduce

noise, the error estimation circuitry must have a small bandwidth, which leads to long convergence times. This tradeoff becomes more severe as the PLL's targeted jitter performance is improved.

A reference frequency-doubling (RFD) technique is presented in this paper which is immune to crystal oscillator duty-cycle error, so it is not subject to the speed-accuracy tradeoff of the prior solutions. It is presented and analyzed in the context of a delta-sigma frequency-to-digital converter (ΔΣ-FDC) based PLL configured to achieve 75 fs rms jitter. Behavioral simulations with nonideal circuit parameters extracted from simulations of transistor-level PLL circuit blocks implemented in Global Foundries 22FDX 22 nm CMOS technology show that the worst-case convergence time is 412 reference cycles. This is at least 10× faster than that of the prior art with comparable initial duty-cycle errors and jitter. The presented ΔΣ-FDC also includes a modified gain calibration technique and achieves reduced PFD and ADC spans after locking relative to comparable prior ΔΣ-FDCs [21], [22].

II. REFERENCE FREQUENCY-DOUBLING IN PLLs

Fig. 2.1(a) shows a top-level block diagram of a generic fractional-N digital PLL where the reference signal, $v_{ref}(t)$, is generated by an FD, so the reference frequency, f_{ref}, is double that of the crystal frequency, $f_{crystal}$. The PLL is designed to generate a periodic output waveform, $v_{PLL}(t)$, with frequency $f_{PLL} = (N+\alpha)f_{ref}$ where N is a positive integer and $-½ \leq \alpha \leq ½$. It consists of a phase-error to digital converter (PEDC), a digital loop filter (DLF), and a digitally controlled oscillator (DCO). The PEDC output, $p[n]$, is a quantized measure of the PLL's phase error and the DCO's frequency control sequence, $d[n]$, is a lowpass filtered version of $p[n]$.

The PEDC in many digital PLLs incorporates an MMD and a PFD as in Fig. 2.1(b). The nth and (n+1)th rising edges of the MMD output, $v_{div}(t)$, are separated by $N-v[n]$ DCO periods, where $v[n]$ is an integer-value sequence generated within the PEDC. The PFD output pulse width, which is equal to the time-difference between the rising edges of $v_{ref}(t)$ and $v_{div}(t)$, is measured and quantized by the phase-error measurement and quantization (PEMQ) circuitry. The sequence $v[n]$ can be generated by a digital re-quantizer such as in [23], [24], [25], and [26] or by linearly filtering $p[n]$ as in [21], [27], [28], [29], and [30]. In each case, the PLL settles such that the mean of $v[n]$ is $-\alpha$, so the PLL's mean output frequency is $(N+\alpha)f_{ref}$.

Fig. 2.1(c) shows the details of a widely-used FD. It consists of an inverter-based delay-line and an XOR-gate arranged such that both the rising and falling edges of its input signal, $v_{crystal}(t)$, cause rising edges in its output signal, $v_{ref}(t)$. Usually, it is mainly the white portion of the phase noise of a crystal oscillator that contributes to a PLL's output phase noise as the crystal's high quality factor relegates other noise to a very low bandwidth. Consequently, to the extent that the FD is ideal, the jitter of $v_{ref}(t)$ is well-modeled as a white sequence and, as shown in [31], the phase noise power spectral density (PSD) of $v_{ref}(t)$ in this case is 3 dB higher than that of $v_{crystal}(t)$. Had the FD not been used in the system of Fig. 2.1, both N and α would have had to be doubled to achieve the same PLL output frequency, and as the PLL's phase noise transfer function from $v_{ref}(t)$ is scaled by $(N+\alpha)^2$, it follows that the FD provides a net 3 dB reduction in the crystal oscillator's contribution to the PLL's phase noise.

The FD also reduces the PEDC's contribution to the PLL's phase noise. The PEDC's measurement noise is a combination of white and highpass shaped components, depending on PEDC's design. As the noise components are essentially sampled at a rate of f_{ref}, doubling f_{ref}

for a given PLL bandwidth reduces the contribution to the PLL's output phase noise of all white and first-order highpass shaped noise components by 3 dB and 9 dB, respectively [7].

Unfortunately, practical FDs, including that shown in Fig. 2.1(c), introduce deterministic jitter when the duty cycle of $v_{\text{crystal}}(t)$ is not exactly 50%, and conventional PLLs are highly sensitive to such jitter. If the crystal oscillator's duty cycle were exactly 50%, the time of the nth rising edge of $v_{\text{ref}}(t)$ could be written as $t_n = nT_{\text{ref}} - j_{\text{ref}}[n]$, where $j_{\text{ref}}[n]$ is the reference signal's jitter [31]. As illustrated in Fig. 2.1(c), when the duty-cycle of $v_{\text{crystal}}(t)$ deviates from 50%, the t_n values are further displaced relative to their ideal values by an alternating error sequence, i.e.,

$$t_n = nT_{\text{ref}} - j_{\text{ref}}[n] + d_e[n], \tag{1}$$

where

$$d_e[n] = \Delta T(-1)^n \quad \text{with} \quad \Delta T = \left(\frac{D}{100} - 0.5\right) T_{\text{ref}}, \tag{2}$$

and D is the crystal oscillator's duty cycle in percent. In practical crystal oscillators, D typically deviates from its ideal value of 50% by anywhere between 5 and 10 percentage points across PVT variations [8], [9], [10], [11], [12].

In a conventional PLL, such duty cycle errors deteriorate the performance of the PEDC circuitry and introduce a large $f_{\text{ref}}/2$ spur at the PLL's output. As $p[n]$ is a quantized measure of the time-difference between the rising edges of $v_{\text{ref}}(t)$ and $v_{\text{div}}(t)$, it contains a term proportional to $d_e[n]$ which causes the above-mentioned $f_{\text{ref}}/2$ spur. The $f_{\text{ref}}/2$ spur could be perfectly removed via a digital notch filter, but this would not eliminate the presence of $d_e[n]$ at the PEMQ's input. For typical values of D, $d_e[n]$ is much larger than $j_{\text{ref}}[n]$ in (1), so the need to accommodate $d_e[n]$

drastically increases the dynamic range, linearity, and noise performance required of the PEMQ.

For example, suppose a 76.8 MHz crystal oscillator with 5% duty-cycle error is used in a PLL to generate a 10 GHz output waveform, and $v[n]$ is generated by a second-order delta-sigma modulator. In the absence of duty-cycle error, the PFD's output nominal span is $2T_{\mathrm{PLL}}$ [32]. It can be deduced from (1) and (2) that a 5% duty-cycle error increases the PFD output span by a factor of 4.25. In a TDC-based PLL, such as presented in [23], this would require two additional bits of TDC dynamic range which would typically quadruple the TDC's power consumption. An increase in the TDC's dynamic range would also degrade the TDC's linearity and, hence, the PLL's spurious tone performance. In a bang-bang PLL, such as presented in [24], the alternating $d_e[n]$ error would push the bang-bang phase-detector far away from its optimal operating point and the PLL may even fail to lock as the bang-bang phase detector's effective gain would be very small [11], [33]. In a charge-pump (CP) FDC-based PLL, such as presented in [21], the increase in the PFD output span would increase the thermal and flicker CP noise contributions by approximately 6.3 and 12.6 dB, respectively. For a 100 fs rms jitter PLL design with a CP jitter contribution of 50 fs, the increase in the CP's white component alone would increase the PLL's jitter by 35%.

To enable high-performance PLLs with sub-100fs rms jitter, spurious tones below −80 dBc, and practical PEMQ performance requirements, either a scheme to reduce D or a scheme to cancel $d_e[n]$ prior to the PEMQ must be employed.

III. PROPOSED RFD TECHNIQUE QUALITATIVE DESCRIPTION

Fig. 2.2 shows a top-level block diagram of the proposed scheme. It is a modified version of the system of Fig. 2.1 in which the PEDC includes an $f_{ref}/2$ resonator in its forward path, there is an $f_{ref}/2$ notch filter between the PEDC and the DLF, and the MMD control word, $v[n]$, is generated by adding $-\alpha$ to a digitally filtered version of $p[n]$ and re-quantizing the result. The transfer function, $F(z)$, of the filter applied to $p[n]$, and the transfer function of the $f_{ref}/2$ resonator are designed such that $p[n]$ is a measure of the PLL's phase-error as in conventional digital PLL architectures, and such that that the PEDC is stable.

The proposed technique can be understood qualitatively as follows. The resonator has an infinite gain at $f_{ref}/2$, so any $f_{ref}/2$ spur at its input would cause its output to grow without bound. However, as the system is stable by design, the output of the resonator must be bounded, so its input must not contain an $f_{ref}/2$ tone. The PEMQ circuitry does not introduce a zero at $f_{ref}/2$ by design, so it follows that the PFD's output must also be free of any $f_{ref}/2$ spur. Consequently, the system must settle such that the times of the MMD rising edges, defined as τ_n for $n = 0, 1, 2, \ldots$, contain a component that is exactly equal to the duty-cycle error sequence, $d_e[n]$, in the reference path, which implies that $d_e[n]$ is perfectly canceled at the PFD's output.

For the MMD output edges to contain a component equal to $d_e[n]$, $p[n]$ must contain a term proportional to $d_e[n]$. The $f_{ref}/2$ notch filter following the PEDC removes this term, thereby preventing it from causing an $f_{ref}/2$ spur in the PLL's output waveform. The effects of the notch filter on the PLL's noise performance and loop dynamics are negligible because the PLL's loop bandwidth is generally much smaller than f_{ref} [32].

As proven in the next section, the proposed technique is free of convergence bias and, in contrast to the techniques presented in [11], [12], [18], [19], and [20], is not subject to the fundamental LMS loop speed-accuracy tradeoff. Furthermore, the PFD outputs only depend on the rising edges of $v_{ref}(t)$, so jitter on the falling edges of $v_{ref}(t)$, which arises primarily from noise introduced by the FD's inverter-based delay chain (Fig. 2.1(c)), does not degrade the PLL's phase noise. Therefore, in contrast to the techniques presented in [13], [14], [15], [16], and [17], there is no additional noise or power consumption penalty associated with adding delay lines in the reference signal path.

If $d_e[n]$ were measured in the analog domain or if the divider or reference edges were shifted in the analog domain by controlling a delay line, the technique would be subject to inaccuracies from nonideal analog circuit behavior. Instead, the proposed technique avoids such inaccuracies by cancelling $d_e[n]$ precisely with a digital-domain feedback path through the MMD via $v[n]$. In principle, the technique can be applied to any digital PLL of the form shown in Fig. 2.1 by adding such a feedback path through the MMD input, $v[n]$. However, FDC-PLLs already contain a feedback path through the MMD to which the technique can be added, so applying the technique to an FDC-PLL requires fewer modifications than applying it to other types of PLLs [21], [27], [29].

IV. PROPOSED FDC-PLL ARCHITECTURE

A. *System Description*

Fig. 2.3(a) shows a top-level block diagram of the proposed FDC-PLL architecture. It has the form of the generic digital PLL in Fig. 2.1(a), but the PEDC is implemented as a cascade

of a second-order ΔΣ-FDC and an accumulator, as in [21], [27], and [29], and a $1+z^{-1}$ block precedes the DLF. As explained shortly, the ΔΣ-FDC is also modified relative to prior ΔΣ-FDCs to incorporate the RFD technique described qualitatively in Section III.

As proven in Section IV-B and Appendix A, the ΔΣ-FDC's output, $r[n]$, is a measure of the PLL's frequency error plus a component proportional to $d_e[n]-d_e[n-1]$. The accumulator following the ΔΣ-FDC performs frequency-to-phase conversion, so its output, $p[n]$, contains terms proportional to the PLL's phase-error and $d_e[n]$. The subsequent $1+z^{-1}$ block prior to the DLF plays the role of the notch filter in Fig. 2.2 as it has a zero at $f_{ref}/2$. The output of the DLF, $d[n]$, is latched into the DCO on each rising edge of $v_{ref}(t)$ such that the DCO's instantaneous frequency during each time interval $t_n \le t < t_{n+1}$ is

$$f_{PLL}(t) = f_c + K_{DCO} d[n-1] + \psi_{DCO}(t), \tag{3}$$

where f_c is the nominal center frequency of the DCO in Hz, K_{DCO} is the DCO gain in Hz, and $\psi_{DCO}(t)$ is the DCO's instantaneous frequency error in Hz [21].

The proposed ΔΣ-FDC shown in Fig. 2.3(b) is an extension of those presented in [21] and [22] that includes the proposed RFD technique. It also includes a modified gain calibration technique, as explained in Section IV-D, and achieves reduced PFD and ADC spans after locking relative to prior ΔΣ-FDCs, as explained in Section IV-E. The ΔΣ-FDC consists of an MMD, a PFD, a CP, an ADC, and a ΔΣ-FDC digital block. The $1/(1+z^{-1})^2$ and $z^{-1}(2-z^{-2})$ transfer functions in the ΔΣ-FDC's digital block play the roles of the $f_{ref}/2$ resonator and $F(z)$ shown in Fig. 2.2, respectively.

The PFD and CP are comparable to those in analog PLLs [32]. Ideally, during the nth reference period, the PFD causes the CP to output a current pulse with a width of $|\tau_n - t_n|$, and

nominal amplitudes of I_{CP} when $t_n < \tau_n$ and $-I_{CP}$ when $\tau_n < t_n$, where τ_n as the time of the nth rising edge of $v_{div}(t)$ and, as mentioned previously, t_n is the time of the nth rising edge of $v_{ref}(t)$. The CP current is integrated by the capacitor, C, so each CP output pulse ideally changes the voltage across the capacitor by $I_{CP}(\tau_n - t_n)/C$ volts.

As in prior ΔΣ-FDCs, the integer-valued MMD control sequence, $v[n]$, is generated as part of the feedback loop within the ΔΣ-FDC. It is a quantized version of $r_F[n]-\alpha$, where $r_F[n]$ is the result of filtering the output of the ΔΣ-FDC by $F(z)$ as shown in Fig. 2.3(b). The quantization is performed by the block labeled Q_C, which is an implementation of the second-order digital ΔΣ modulator shown in Fig. 2.3(c).

The ΔΣ-FDC's B-bit ADC samples the CP output voltage at the rising edges of $v_{samp}(t)$, which is a delayed version of $v_{ref}(t)$. The ADC's output, $a[n]$, is interpreted as a fixed-point two's complement number with $B-F$ and F integer and fractional bits, respectively. Each integer step of the ADC output corresponds to an ADC input step of Δ volts, so the ADC output sequence is interpreted as having a minimum step-size of $2^{-F}\Delta$ and an integer step-size of Δ.

The ADC's output is multiplied by the FDC's gain calibration loop output, $\hat{g}[n]$, and the sequence $e_{qc}[n-1]$ is added to the result to cancel quantization error introduced by Q_C that would otherwise degrade the PLL's phase noise. As explained in Section IV-E, an additional benefit of this quantization-error cancellation (QNC) technique is that it reduces the PFD and ADC spans compared to those in [21] when the PLL is locked.

The multiplication by $\hat{g}[n]$ corrects for gain error incurred in the ΔΣ-FDC's forward path such as can result from deviations of I_{CP}, C, and Δ from their nominal values. As explained in [21], ΔΣ-FDCs are not generally sensitive to such gain errors in terms of their input-output

transfer functions. However, for a low-jitter PLL, the gain error must be low enough for QNC to sufficiently suppress $e_{qc}[n]$. For instance, in the PLL design example presented in Section V, the gain error must be 1% or less for the leaked component of $e_{qc}[n]$ at the PLL's output to be at least 10 dB lower than any other noise source so as to negligibly degrade the PLL's phase noise.

The FDC Gain Calibration block together with the $\hat{g}[n]$ multiplier and QNC adder implement a sign-LMS-like loop with loop gain K, reference sequence $e_{qc}[n]$, and error sequence $c[n]$. The FDC gain calibration technique can be qualitatively understood as follows. If $\hat{g}[n]$ is larger or smaller than its ideal value, then $e_{qc}[n]$ is not perfectly canceled at the QNC adder and $c[n]$ contains a term proportional to $-e_{qc}[n-1]$ or $e_{qc}[n-1]$, respectively. The FDC Gain Calibration block multiplies $c[n]$ by the sign of $e_{qc}[n-1]$ and accumulates the result, so the term proportional to $-e_{qc}[n-1]$ or $e_{qc}[n-1]$ in $c[n]$ respectively decreases or increases the accumulator output by $|e_{qc}[n-1]|$. As all other terms in $c[n]$ have zero-mean, this causes the $\hat{g}[n]$ to be reduced or increased until $\hat{g}[n]$ reaches its ideal value aside from zero-mean fluctuations caused by noise. As with other LMS-like loops, reducing the magnitude of the loop gain, K, reduces the noise fluctuations at the expense of convergence rate.

B. ΔΣ-*FDC Linearized Model*

As proven in Appendix A, the proposed ΔΣ-FDC has a linearized model as shown in Fig. 2.4(a) for the case where the FDC gain calibration loop has converged such that $\hat{g}[n]$ can be approximated as a constant value, $\hat{g}_{\mathrm{FDC}}$. In Fig. 2.4(a), $\theta_{\mathrm{ref}}[n]$ is the reference phase noise, in cycles, at time t_n, $\theta_{\mathrm{PLL}}[n]$ is the PLL's phase noise, in cycles, at time τ_n, and $e_{\mathrm{CP}}[n]$ and $e_{\mathrm{ADC}}[n]$ represent error introduced by the CP and ADC, respectively.

The nominal values of I_{CP}, C, and Δ are chosen to satisfy $T_{PLL}I_{CP}/C\Delta = 1$, because, as can be deduced from Fig. 2.4(a), this with $\hat{g}_{FDC} = 1$ causes the contribution of $e_{ADC}[n]$ to $r[n]$ to have the desired second-order highpass spectral shape, and causes the adder with output $c[n]$ to perfectly cancel $e_{qc}[n–1]$ in $r[n]$. In practice, however, I_{CP}, C, and Δ, deviate from their nominal values, so, as explained in Section IV-D, the output of the FDC's gain calibration loop converges such that the $\Delta\Sigma$-FDC's forward path gain is unity, i.e.,

$$T_{\mathrm{PLL}} \frac{I_{CP}}{C} \frac{1}{\Delta} \hat{g}_{\mathrm{FDC}} = 1. \tag{4}$$

Fig. 2.4(a) and (4) imply that the $\Delta\Sigma$-FDC's output is

$$r[n] = -\left(\theta_{\mathrm{PLL}}[n] - \theta_{\mathrm{PLL}}[n-1]\right) + e_{\mathrm{FDC}}[n] - e_{\mathrm{FDC}}[n-1] + f_{\mathrm{PLL}}\left(d_e[n] - d_e[n-1]\right), \tag{5}$$

where

$$e_{\mathrm{FDC}}[n] = (N+\alpha)\theta_{\mathrm{ref}}[n] + \frac{\hat{g}_{\mathrm{FDC}}}{\Delta} e_{CP}[n] + \hat{g}_{\mathrm{FDC}}\left(e_{\mathrm{ADC}}[n] - e_{\mathrm{ADC}}[n-1]\right) \tag{6}$$

represents error introduced by the reference signal and the $\Delta\Sigma$-FDC. As shown in Fig. 2.3(a), $r[n]$ is accumulated to generate $p[n]$, so (5) implies

$$p[n] = -\theta_{\mathrm{PLL}}[n] + e_{\mathrm{FDC}}[n] + f_{\mathrm{PLL}} d_e[n]. \tag{7}$$

It follows from Fig. 2.4(a) that the transfer function from $d_e[n]$ to $u[n]$, where $u[n]$ is defined as $\tau_n - t_n$, is

$$(1-z^{-1})^2(1+z^{-1})^2. \tag{8}$$

Hence, the transfer function from $d_e[n]$ to $u[n]$ has a pair of zeros at $z = -1$, so $d_e[n]$, which is proportional to $(-1)^n$, does not appear at the PFD's output, which proves the corresponding result presented and explained qualitatively in Section III.

Although the FDC gain calibration loop provides the benefit outlined in Section IV-A, it is not necessary for the RFD technique to function properly. If the ΔΣ-FDC's loop gain is left uncalibrated such that

$$T_{\mathrm{PLL}} \frac{I_{CP}}{C} \frac{1}{\Delta} \hat{g}_{\mathrm{FDC}} = 1 + g \,, \tag{9}$$

where g is the gain-error, it follows from Fig. 2.4(a) and (4) that the transfer function from $d_e[n]$ to $u[n]$ becomes:

$$\frac{(1-z^{-1})^2 (1+z^{-1})^2}{1+2gz^{-2}-gz^{-4}}. \tag{10}$$

Therefore, the transfer function has a pair of zeros at $z = -1$ like (8), so $d_e[n]$ does not appear at the PFD's output even when the ΔΣ-FDC has a gain error.

The gain error does introduce poles, though, which slightly increases the initial $d_e[n]$ settling time. After a cold-start, the PLL's dynamics are nonlinear so the linearized model of Fig. 2.4(a) does not apply and the ΔΣ-FDC's gain error has little effect. Once all analog nodes and digital registers stop clipping and remain within their linear operating regions, the ΔΣ-FDC linearized model becomes applicable and (10) can be used to evaluate the settling of the $d_e[n]$ component of $u[n]$. Without loss of generality, suppose the ΔΣ-FDC linearized model becomes applicable at $n = 0$. Convolving $d_e[n]$ with the inverse z-transform of (10) shows that the component of $u[n]$ corresponding to $d_e[n]$ is

$$h_{de}[n] = \begin{cases} \dfrac{\Delta T}{p_1 - p_2} \left\{ (p_1 - 1)\, p_1^{n/2} - (p_2 - 1)\, p_2^{n/2} \right\} & n = \text{even} \\ -h_{de}[n-1] & n = \text{odd} \end{cases} \tag{11}$$

for $n = 0, 1, 2, \ldots$, and

$$p_{1,2} = -g \pm \sqrt{g^2 + g}\ . \tag{12}$$

For gain errors bounded in magnitude by 20% (i.e., |g| < 0.2), (11) implies that $h_{de}[n]$ decays with time and its magnitude becomes equivalent to a duty-cycle error less than 0.005% in at most 22 reference cycles. For the PLL parameters used in the design presented in Section V, this level corresponds to an $f_{\text{ref}}/2$ spur less than −80 dBc and adds less than 0.5% to the CP and ADC nominal spans. Even for a gain error of 30%, which is far larger than would be expected in practice, a residual duty-cycle error of 0.005% is reached in 102 reference cycles.

C. *PLL Linearized Phase Noise Model*

When the PLL is locked, its ideal output frequency is $(N+\alpha)f_{\text{ref}}$, so it follows from (3) that the DCO input sequence can be written as $d[n] = [(N+\alpha)f_{\text{ref}} - f_{\text{c}}]/K_{\text{DCO}} + f[n]$, where $f[n]$ is the component of $d[n]$ arising from noise. This with (3) implies that during each time interval $t_n \le t < t_{n+1}$, the PLL's instantaneous frequency error in Hz can be expressed as

$$\psi_{\text{PLL}}(t) = K_{\text{DCO}}\, f[n-1] + \psi_{\text{DCO}}(t)\,. \tag{13}$$

Phase is the integral of frequency, so integrating (13) from t_0 to t, for $t_n \le t < t_{n+1}$ with $n \ge 0$ gives

$$\theta_{\text{PLL}}(t) = \theta_{\text{DCO}}(t) + K_{\text{DCO}}\left(t - t_n\right) f[n-1] + K_{\text{DCO}} \sum_{k=-1}^{n-2} \left(t_{k+2} - t_{k+1}\right) f[k], \tag{14}$$

where it has been assumed without loss of generality that $\theta_{\text{PLL}}(t_n) = 0$ and $\theta_{\text{DCO}}(t_n) = 0$. Typical reference oscillators have high spectral purity, so (1) implies

$$t_{k+2} - t_{k+1} \cong T_{\text{ref}} + \Delta T(-1)^n\ . \tag{15}$$

Substituting this into (14) gives

$$\theta_{\mathrm{PLL}}(t) \cong \theta_{\mathrm{DCO}}(t) + K_{\mathrm{DCO}}\left(t - t_n\right) f[n-1] + \theta_{\mathrm{loop}}[n] + K_{\mathrm{DCO}}\Delta T \sum_{k=-1}^{n-2} (-1)^k f[k], \tag{16}$$

where

$$\theta_{\mathrm{loop}}[n] = K_{\mathrm{DCO}} T_{\mathrm{ref}} \sum_{k=-1}^{n-2} f[k]\,. \tag{17}$$

The bandwidth of a PLL is generally much smaller than f_{ref} and $|\tau_n - t_n|$ is less than a few DCO periods, so

$$\theta_{\mathrm{PLL}}(\tau_n) \cong \theta_{\mathrm{PLL}}(t_n)\,. \tag{18}$$

Therefore, (16) implies

$$\theta_{\mathrm{PLL}}[n] \cong \theta_{\mathrm{DCO}}[n] + \theta_{\mathrm{loop}}[n] + K_{\mathrm{DCO}}\Delta T \sum_{k=-1}^{n-2} (-1)^k f[k]\,, \tag{19}$$

where $\theta_{\mathrm{DCO}}[n] = \theta_{\mathrm{DCO}}(t_n)$ and $\theta_{\mathrm{PLL}}[n] = \theta_{\mathrm{PLL}}(t_n)$.

Equations (16) and (19) are linear difference equations, but they are not time-invariant when the crystal oscillator has a non-50% duty cycle because of the $(-1)^k f[n]$ terms. As shown in Appendix B, they give rise to the linear time-varying (LTV) PLL phase noise model shown in Fig. 2.5, where $L(z)$ is the transfer function from $p[n]$ to $d[n]$ in Fig. 2.3(a), each FOH block is a first-order hold (FOH) interpolator, and the expressions for $H_{\mathrm{err}}(j\omega)$ and $H^a_{\mathrm{err}}(j\omega)$ are given in Appendix B. If the duty cycle of the crystal oscillator were exactly 50%, then $H_{\mathrm{err}}(j\omega)$ and $H^a_{\mathrm{err}}(j\omega)$ would equal 1 and 0, respectively, and the phase noise model would reduce to the LTI phase noise model presented in [21] despite the modifications of the PLL presented in this paper.

The output of each FOH interpolator is a continuous-time waveform given by

$$\sum_{n=0}^{\infty} s[n] h_{tri}\left(t - nT_{\text{ref}} - t_0\right), \tag{20}$$

where

$$h_{\text{tri}}(t) = \begin{cases} 1 - |t|/T_{\text{ref}}, & \text{if } |t| < T_{\text{ref}}, \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

and $s[n]$ is a dummy variable which represents the FOH interpolator's input sequence. The continuous-time Fourier transform (CTFT) of (20) is

$$T_{\text{ref}} H(\omega) S\left(e^{j\omega T_{\text{ref}}}\right) \text{ where } H(\omega) = \left[\frac{\sin\left(\omega T_{\text{ref}}/2\right)}{\omega T_{\text{ref}}/2}\right]^2, \tag{22}$$

and $S(e^{j\omega T\text{ref}})$ is the discrete-time Fourier transform (DTFT) of $s[n]$ [34]. Therefore, the CTFT of the sum of $H_{\text{err}}(j\omega)$ and $H^a_{\text{err}}(j\omega)$ outputs in Fig. 2.5 is

$$X\left(e^{j\omega T_{ref}}\right) C(\omega) + X\left(e^{j\left(\omega T_{ref} - \pi\right)}\right) C^a(\omega), \tag{23}$$

where $X(z)$ is the z-transform of $x[n] = -\theta_{\text{DCO}}[n] + e_{\text{FDC}}[n]$,

$$C(\omega) = T_{ref} G\left(e^{j\omega T_{ref}}\right) H(\omega) H_{err}(j\omega), \tag{24}$$

$$C^a(\omega) = T_{ref} G\left(e^{j\left(\omega T_{ref} - \pi\right)}\right) H(\omega) H^a{}_{err}(j\omega), \tag{25}$$

$$G(z) = \frac{T(z)}{1 + T(z)}, \quad \text{and} \quad T(z) = z^{-2} L(z) \frac{K_{\text{DCO}} T_{\text{ref}}}{1 - z^{-1}}. \tag{26}$$

Figures 2.6(a) and 2.6(b) show the magnitudes in dB of $C(\omega)$ and $C^q(\omega)$ versus frequency in Hz for the PLL design example presented in Section V with loop bandwidths of 280 kHz and 1.3 MHz. The results imply that the $H^a_{\text{err}}(j\omega)$ path in the phase noise model has little effect for the design example. The reason is that $H^a_{\text{err}}(j\omega)$ is highly attenuated over the bandwidth of its input signal. Furthermore, as illustrated in Fig. 2.6(c), $H_{\text{err}}(j\omega) \cong 1$ to a high

degree of accuracy for the design example up to frequencies well past the PLL's bandwidth. Consequently, the phase noise model is well-approximated for the design example by the system of Fig. 2.5 with $H_{\text{err}}(j\omega) = 1$ and the $H^a_{\text{err}}(j\omega)$ path neglected.

To the extent that the PLL's noise sources can be modeled as uncorrelated zero-mean wide-sense stationary random processes, it follows that the PSD of $\theta_{\text{PLL}}(t)$ is the superposition of the PSDs of the individual sources. Hence, the two-sided PSD of $\theta_{\text{PLL}}(t)$ is:

$$S_{\theta_{\text{PLL}}}(f) = S_{\theta_{\text{PLL}}}(f)\Big|_{\text{ref}} + S_{\theta_{\text{PLL}}}(f)\Big|_{\text{CP}} + S_{\theta_{\text{PLL}}}(f)\Big|_{\text{ADC}} + S_{\theta_{\text{PLL}}}(f)\Big|_{\text{DCO}}, \tag{27}$$

where the terms on the right hand side of (27) are the two-sided PSDs of the reference source, CP, ADC, and DCO noise contributions to the PLL's output. Following the same reasoning in [21], Table 2.1 summarizes the contribution of the different sources above for the PLL phase noise model with $H_{\text{err}}(j\omega) = 1$ and $H^a_{\text{err}}(j\omega) = 0$.

D. ΔΣ-*FDC Gain Calibration Details*

Fig. 2.4(b) shows the portion of the ΔΣ-FDC's behavioral model connecting the ADC's output, $a[n]$, and the resonator's input, $c[n]$, with the FDC gain calibration loop details added. As explained in Sections IV-A and IV-B, the objective of the FDC gain calibration loop is to cause $\hat{g}[n]$ to converge to a constant, $\hat{g}_{\text{FDC}}$, aside from zero-mean error, that satisfies (4). Therefore (4) implies that $\hat{g}[n]$ can be written as

$$\hat{g}[n] = \frac{1}{A} + \varepsilon[n], \text{ with } A = T_{\text{PLL}} \frac{I_{CP}}{C} \frac{1}{\Delta}, \tag{28}$$

where $\varepsilon[n]$ is the FDC gain calibration error.

It follows from Fig. 2.4(b) that $\hat{g}[n] = \hat{g}[n-1] + Ks[n-1]$, so (28) implies

$$\varepsilon[n] = \varepsilon[n-1] + Ks[n-1]. \tag{29}$$

This and the linearity of the expectation operator implies

$$\overline{\varepsilon}[n] = \overline{\varepsilon}[n-1] + K\overline{s}[n-1], \tag{30}$$

where $\overline{s}[n]$ and $\overline{\varepsilon}[n]$ are the expectations of $s[n]$ and $\varepsilon[n]$, respectively. To the extent that the gain-error does not deteriorate the self-dithering property of delta-sigma modulators to a great extent, $e_{qc}[n]$ is well-approximated as independent of all other random variables and uniformly distributed between –1/2 and 1/2. A nearly identical analysis to that presented in [35] shows that

$$\overline{s}[n] = -A\overline{\varepsilon}[n] \cdot E\left\{\left|e_{qc}[n-1]\right|\right\}. \tag{31}$$

The uniform distribution of $e_{qc}[n]$ between –1/2 and 1/2 implies that $|e_{qc}[n]|$ is uniformly distributed between 0 and 1/2, so it follows that

$$\overline{\varepsilon}[n] = \overline{\varepsilon}[n-1]\left(1 - \frac{AK}{4}\right). \tag{32}$$

Recursively substituting (32) in itself yields

$$\overline{\varepsilon}[n] \cong \varepsilon[0]\left(1 - \frac{AK}{4}\right)^n, \tag{33}$$

hence, $\overline{\varepsilon}[n]$ converges to zero, so $\hat{g}[n]$ converges to $\hat{g}_{\mathrm{FDC}}$ aside from zero-mean error, as $n \to \infty$, provided $0 < K < 4/A$.

As with any LMS-like loop, the FDC gain calibration technique is subject to a convergence speed versus accuracy trade-off; increasing K increases the convergence rate, but it also increases the power of $\varepsilon[n]$. However, as implied by the signal processing operations shown in Fig. 2.3, the contribution of $\varepsilon[n]$ to the PEDC output, $p[n]$, is $\varepsilon[n]-\varepsilon[n-1]$, so the FDC gain calibration error is subjected to first-order highpass shaping. This reduces its contribution

to the PLL phase noise, thereby relaxing the convergence rate versus accuracy tradeoff relative to the FDC gain calibration technique presented in [35].

E. *Additional ΔΣ-FDC Properties*

In prior CP-based ΔΣ-FDCs, such as those presented in [21] and [22], the $v[n]$ input to the MMD is $2y[n]-y[n-1]$, where $y[n]$ denotes the integer portion of the output of either the ADC or, if FDC gain calibration is implemented, the FDC gain calibration multiplier following the ADC. Ideally, the time at which the CP output pulse terminates during the nth reference period is the larger of t_n and τ_n, but, in practice, the CP takes time to settle and the ADC then takes time to perform a conversion, so $y[n]$ is not available until well after the start of the reference period. Once $y[n]$ is available, $v[n]$ must be computed and loaded into the MMD early enough that the time of the MMD's next rising output edge, τ_{n+1}, is $N-v[n]$ DCO periods after that of the MMD's prior rising output edge, τ_n. These timing constraints can be tight, especially for high reference frequencies.

In contrast, the proposed ΔΣ-FDC (Fig. 2.3(b)) has an extra reference period delay between the ADC output and $v[n]$ relative to prior CP-based ΔΣ-FDCs because of the delay through $F(z)$. Although the QNC adder, resonator, and $F(z)$ block represent more digital operations than are needed to just compute $2y[n]-y[n-1]$ in prior ΔΣ-FDCs, they can be performed in a small fraction of a reference period in typical CMOS technology, so the timing constraints of the proposed ΔΣ-FDC are significantly more relaxed than those of prior CP-based ΔΣ-FDCs.

However, the extra feedback delay through $F(z)$ causes the proposed ΔΣ-FDC to have a smaller maximum input frequency range than that of prior CP-based ΔΣ-FDCs with the same

ADC. This is because the deviation of the average DCO frequency per reference period relative to its ideal value of $(N+\alpha)f_{\text{ref}}$ affects the ADC output through a transfer function of $(1+z^{-1})^2/f_{ref}$ in the proposed ΔΣ-FDC but of just $1/f_{ref}$ in prior CP-based ADCs. The issue mainly affects the PLL during the locking process because, once the PLL's output frequency has converged to $(N+\alpha)f_{\text{ref}}$, the ADC output sequence is dominated by coarse quantization error from Q_C, the span of which is not affected by the extra delay in $F(z)$. Therefore, the required ADC span is higher while the PLL locks than it is once the PLL finishes locking. In the design example presented in the next section, this issue is addressed via a SAR ADC that provides 7 bits of resolution while the PLL locks and then reduces its resolution to 6 bits to save power once the PLL has locked.

In contrast, three other modifications of the proposed ΔΣ-FDC—performing QNC within the ΔΣ-FDC, and subtracting α and performing the Q_C quantization within the ΔΣ-FDC's feedback path—act to relax the required ADC input range compared to prior CP-based ΔΣ-FDCs. They do so by reducing the contribution of coarse quantization error, $e_{qc}[n]$, to the ADC's input, which makes the biggest difference after the PLL locks when the ADC span is dominated by $e_{qc}[n]$. Fig. 2.3(c) implies that $|e_{qc}[n]| \leq ½$, and Fig. 2.4(a) implies that the transfer function from $e_{qc}[n]$ to the ADC's input is nominally z^{-1}, so the ADC in the proposed ΔΣ-FDC requires only one integer ADC step to accommodate coarse quantization error. In contrast, each of the ΔΣ-FDC's presented in [21] and [22] require three integer ADC steps to accommodate coarse quantization error. For example, the ΔΣ-FDCs in [21] and [22] each require a total input range of 4 integer ADC steps to achieve a frequency acquisition range of f_{ref}, whereas the proposed

ΔΣ-FDC requires total input ranges of 5 and 1.25 integer ADC steps before and after the PLL locks, respectively.

Similar reasoning shows that the modifications also reduce the average duration of the CP output pulses relative to prior CP-based ΔΣ-FDCs. In the proposed ΔΣ-FDC, the average CP pulse duration is reduced by a factor of 2.6 relative to prior CP-based ΔΣ-FDCs. This corresponds to a reduction in the CP's thermal and flicker noise contributions to the PLL's phase noise by 4.1 dB and 8.2 dB, respectively [32]. The thermal noise reduction alone reduces the CP's contribution to the PLL's jitter by 37% for a given PLL bandwidth and output frequency.

V. PLL DESIGN EXAMPLE

This The design example of the proposed PLL presented in this section has $f_{crystal}$ = 76.8 MHz, f_{PLL} = 10 GHz, and an RMS output jitter of 75 fs. Table 2.2 presents the relevant design parameters and noise contributions. The noise contributions and other nonideal circuit behavior, such as CP nonlinearity and component mismatches, were determined via Cadence Spectre simulations of FD, CP, and DCO circuits implemented in the Global Foundries 22-nm CMOS 22FDX process. Parameters that describe the nonideal circuit behavior were extracted from the transistor-level simulations and back-annotated into a custom, C-language, event-driven, bit-exact, behavioral simulator along the lines of those described in [27], [30], [35], and [38]. The events modeled by the behavioral simulator are the rising and falling edges of the crystal oscillator, FD, MMD, PFD and DCO.

The CP circuit incorporates the offset-current linearization technique presented in [36]. To capture the CP's nonlinear behavior, the CP's transistor-level simulated output voltages versus the expected range of PFD output pulse-widths in increments of 1 ps were back-annotated into a look-up table (LUT), which the behavioral simulator uses to calculate each CP output voltage via piecewise linear interpolation between adjacent LUT points. The transistor-level simulation testbench included realistic models for the supply network (including the supply source impedance, routing traces, bond-wires, and decoupling capacitors) to capture the effect of the PFD transitions on the power supply shared with the CP. The offset-current amplitude was set equal to that of the CP and its pulse width was set to 200 ps, which simulations predicted is sufficient for supply ripples to not significantly degrade the CP linearity.

The ΔΣ-FDC's ADC is a 7-bit asynchronous SAR ADC with 2 integer bits and 5 factional bits. The corresponding frequency acquisition range of the ΔΣ-FDC is 30 MHz, which transistor-level simulations suggest is more than sufficient to cover temperature and flicker-noise induced DCO frequency drifts. The standard deviation of the unit capacitor random mismatch is set to two percent in the behavioral model, based on Monte Carlo simulations in Spectre. After locking, the input range of the ADC is such that only 1 integer bit is required as explained in Section IV-E, so the ADC resolution is reduced to 6 bits to save power. Behavioral simulations suggest that a comparator metastability rate of 0.01% is sufficient to not significantly degrade the PLL's performance, which is not difficult to satisfy in practice [37].

All digital operations performed by the behavioral simulator are bit-exact. The bus widths of α, $e_{qc}[n]$, $r[n]$, $v[n]$, $p[n]$, and $d[n]$ are 18, 18, 20, 7, 19, and 16 bits respectively. The

FDC gain calibration accumulator has a bus width of 25 bits, which is truncated to 15-bits to generate $\hat{g}[n]$. The DLF consists of a conventional proportional-integral stage, and one single-pole IIR stage [19], [29], [38]. The transfer function from $p[n]$ to $d[n]$ is given by:

$$L(z) = \left(1 + z^{-1}\right)\left(K_P + K_I \frac{z^{-1}}{1 - z^{-1}}\right)\left(\frac{1 - \lambda}{1 - \lambda z^{-1}}\right), \tag{34}$$

where K_P and K_I are the proportional and integral path gains, respectively, λ is the pole of the IIR stage, and the $1+z^{-1}$ factor represents the $f_{\text{ref}}/2$ notch filter.

Fig. 2.7 shows the PLL's various phase noise spectra. The fractional spurs at integer multiples of 120.56 kHz are from CP nonlinearity. Their total power is just under −56 dBc and their presence increases the jitter from 66 fs to 75 fs. Additional simulations run by the authors indicate that in the absence of the FD, the PLL's total jitter would have been 90 fs.

The convergence time of the proposed RFD technique, in reference cycles, is defined as $N_{\text{conv}} = N_{\text{non-lin}} + N_{\text{lin}}$. The first term, $N_{\text{non-lin}}$, is the number of reference cycles after the DCO's coarse frequency is set to an initial value between $(N+\alpha)f_{ref} - 15 \cdot 10^6$ Hz and $(N+\alpha)f_{ref} + 15 \cdot 10^6$ Hz that are required for the PLL to settle to the point where the ΔΣ-FDC's linearized model (Fig 2.4) holds. The second term, N_{lin}, is the number of reference cycles required for $h_{de}[n]$ in (11) to decay to a magnitude which corresponds to a duty-cycle error less than 0.005%. Plotting (11) shows that N_{lin} increases with the ΔΣ-FDC gain-error and is equal to 22 for the worst-case ΔΣ-FDC gain error of 20%. For simplicity, the worst-case value of $N_{\text{lin}} = 22$ is assumed in the following.

Fig. 2.8 shows a histogram of $N_{\text{non-lin}}$ for 10,000 PLL runs. For each run, the PLL was initialized to have a random crystal oscillator initial phase, a crystal-oscillator duty-cycle

between 40% and 60%, an uncalibrated ΔΣ-FDC gain-error between −20% and 20%, and an initial DCO frequency error between −15 MHz and 15 MHz. Fig. 2.8 and N_{lin} = 22 imply that the maximum and average convergence times are 412 and 157 reference cycles, respectively.

Table 2.3 compares the worst-case convergence time of the design example to the published convergence times of the published prior. It shows that the design example has a significantly lower convergence time than the published prior art even with higher crystal oscillator duty cycle errors and lower jitter. The closest competitor is [12], but its reported convergence time is not directly comparable to that of the design example. As described in [11], behavioral simulation results indicate that the LMS duty-cycle calibration loop converges to a duty cycle error of 6% in 2000 reference cycles, after which the LMS loop bandwidth is reduced followed by an unspecified additional convergence time to prevent the LMS loop noise from degrading the PLL's phase noise. As it was not specified in [11], the corresponding value in Table 2.3 does not include the extra required convergence time. Therefore, the convergence time of the design example is pessimistically at least 5× faster than that the published prior art, but the authors estimate that it is at least 10× faster than that of the prior art with comparable initial duty-cycle errors, comparable PLL jitter, and negligible added LMS loop noise.

To evaluate the convergence speed of the FDC gain calibration loop and compare it with that predicted by (33), $\overline{\varepsilon}[n]$ was simulated by averaging $\varepsilon[n]$ over 1000 PLL simulation runs. As the derivation which led to (33) assumes that the PLL is locked, the FDC gain calibration loop was enabled after the PLL locked to provide a meaningful comparison. Fig. 2.9 shows the simulated and calculated values of $\overline{\varepsilon}[n]$ for an initial ΔΣ-FDC gain error of 20% and different values of the LMS loop gain, K. It shows that the simulated and calculated values of $\overline{\varepsilon}[n]$ are

within 2% of each other and decay to within 1% of the ideal gain, 1/*A*, in less than 750 reference cycles for the value of $K = 2^{-6}$ used in the design example. The 1% error threshold ensures that the leaked $e_{qc}[n]$ component contribution to the PLL's phase noise is 10 dB less than all other noise sources. Behavioral simulations performed by the authors show that similar convergence results are achieved when the FDC gain calibration loop is enabled before the PLL loop starts locking.

APPENDIX A: ΔΣ-FDC LINEARIZED MODEL DERIVATION

As explained in Section IV-A, the CP current is integrated by the capacitor, *C*, and changes the voltage across the capacitor by $I_{CP}(\tau_n - t_n)/C$ volts during the *n*th reference period. Hence, the CP output voltage is

$$v_{CP}[n] = v_{CP}[n-1] + \frac{I_{CP}}{C}(\tau_n - t_n) + e_{CP}[n], \tag{35}$$

where τ_n and t_n are the times of the *n*th rising edges of $v_{div}(t)$ and $v_{ref}(t)$, respectively, and $e_{CP}[n]$ is the noise and distortion added by the CP. As proven in [21],

$$\tau_n = \tau_0 - T_{PLL}\theta_{PLL}[n] + T_{PLL}\sum_{k=1}^{n}\left(N - v[k-1]\right), \tag{36}$$

and it follows from Fig. 2.3(b) and Fig. 2.3(c) that

$$v[n] = -\alpha - r_F[n] + e_{qc}[n] - 2e_{qc}[n-1] + e_{qc}[n-2]. \tag{37}$$

The reference jitter can be written as $j_{ref}[n] = T_{ref}\theta_{ref}(t_n)$, where $\theta_{ref}(t)$ is the reference phase noise in cycles [31]. Consequently, (1)can be re-written as

$$t_n = nT_{ref} - T_{ref}\theta_{ref}(t_n) + d_e[n]. \tag{38}$$

The ADC samples and quantizes the CP output voltage at the rising edges of $v_{\text{samp}}(t)$. The time of the rising edge of $v_{\text{samp}}(t)$ during the nth reference period is greater than both τ_n and t_n and less than both τ_{n+1} and t_{n+1}, and each integer step of the ADC output corresponds to an ADC input step of Δ volts, so the ADC output can be written as

$$a[n] = \frac{1}{\Delta} v_{\text{CP}}[n] + e_{\text{ADC}}[n], \tag{39}$$

where $e_{\text{ADC}}[n]$ is the noise and distortion added by the ADC.

The $\Delta\Sigma$-FDC linearized model shown in Fig. 2.10 follows from (35) through (39) and Fig. 2.3(b) as follows. The shaded blocks in Fig. 2.10 labeled MMD, reference source, and ADC graphically implement (36), (38), and (39), respectively, and those labeled PFD and CP together graphically implement (35). The τ_0 term in (35) is not shown explicitly in the CP block of Fig. 2.10 because it can be interpreted as just contributing an initial condition of $\tau_0 I_{\text{CP}}/C$ to the CP block's accumulator, so it does not affect the $\Delta\Sigma$-FDC linearized model's transfer functions. The forward path blocks and the $z^{-1}(2-z^{-2})$ block within the FDC digital block in Fig. 2.10 are those shown in Fig. 2.3(b) where $\hat{g}_{\text{FDC}}$ is the value to which $\hat{g}[n]$ converges, and the remaining blocks within the FDC digital block in Fig. 2.10 graphically implement (37).

As $T_{\text{ref}} = T_{\text{PLL}}(N+\alpha)$, the portion of the output of the MMD accumulator in Fig. 2.10 corresponding to the $(N+\alpha)T_{\text{ref}}$ component of its input is nT_{ref}. This term cancels the nT_{ref} term introduced by the reference source at the PFD's differencer. Eliminating nT_{ref}, N, and α and rearranging the MMD and Q_{C} portions of Fig. 2.10 results in the linearized model shown in Fig. 2.4(a).

Substituting (17) into (16) gives

$$\theta_{\mathrm{PLL}}(t) = \theta_{\mathrm{DCO}}(t) + K_{\mathrm{DCO}}(t - t_n)\, f[n-1] + \theta_{\mathrm{LTV}}[n] \tag{40}$$

for $t_n \le t < t_{n+1}$, where

$$\theta_{\mathrm{LTV}}[n] = K_{\mathrm{DCO}} T_{\mathrm{ref}} \sum_{k=-1}^{n-2} \left(1 + (-1)^k \mu\right) f[k], \tag{41}$$

and $\mu = 2\Delta T/T_{\mathrm{ref}}$. Equation (40) with $t = t_n$ reduces to $\theta_{\mathrm{PLL}}[n] = \theta_{\mathrm{DCO}}[n] + \theta_{\mathrm{LTV}}[n]$. This with (7) implies that $p[n]$ in Fig. 2.3(a) can be written as $p[n] = -\theta_{\mathrm{DCO}}[n] + e_{\mathrm{FDC}}[n] - \theta_{\mathrm{LTV}}[n]$. As $L(z)$ is the transfer function from $p[n]$ to $d[n]$, the component of $d[n]$ corresponding to noise is $f[n]$, and $p[n]$ is a noise sequence, it follows that $L(z)$ is the transfer function from $p[n]$ to $f[n]$. The portion of Fig. 2.11(a) between the $f[n]$ node and the node labeled $\theta_{\mathrm{LTV}}[n]$ is a graphical representation of (41). It down-samples $f[n]$ into a stream of even-indexed samples scaled by $1+\mu$ and a stream of odd-index samples scaled by $1-\mu$, combines the streams via up-sampling and time-shift operations, and accumulates and scales the result. Together, these observations prove that the output of the feedback loop in Fig. 2.11(a) is indeed $\theta_{\mathrm{LTV}}[n]$.

The second and third terms on the right side of (40) represent a linear interpolation operation between t_n and t_{n+1}, where, for each n, t_n is given by (1) with $d_e[n]$ given by (2). In typical PLLs, the reference source jitter is low enough that its effect on the interpolation operation is negligible [21]. However, for typical levels of duty cycle error, the effect of $d_e[n]$ on the interpolation process is not necessarily negligible. As $d_e[n] = \Delta T(-1)^n$, it follows that

$$\theta_{\mathrm{PLL}}(t) = \theta_{\mathrm{DCO}}(t) + \sum_{n=0}^{\infty} \theta_{\mathrm{LTV}}[2n] h_{tri}^{e}\left(t - nT_{\mathrm{ref}} - \Delta T\right) + \sum_{n=0}^{\infty} \theta_{\mathrm{LTV}}[2n+1] h_{tri}^{o}\left(t - (n+1)T_{\mathrm{ref}} + \Delta T\right), \tag{42}$$

for all $t > 0$, where $h^e_{\text{tri}}(t)$ and $h^o_{\text{tri}}(t)$ are as shown in Fig. 2.11(b). The contributions of the two summations in (42) are illustrated in Fig. 2.12, from which it can be seen that the samples of $\theta_{\text{LTV}}[n]$ are first-order-hold interpolated between times $nT_{\text{ref}} + \Delta T$ and $(n+1)T_{\text{ref}} - \Delta T$ when n is even and between times $nT_{\text{ref}} - \Delta T$ and $(n+1)T_{\text{ref}} + \Delta T$ when n is odd. The portion of Fig. 2.11(a) between the $\theta_{\text{LTV}}[n]$ node and the output is a graphical implementation of (42), wherein the FOH_e and FOH_o interpolators respectively implement (20) with $s[n]$ replaced by the even-index samples of $\theta_{\text{LTV}}[n]$ and $h_{tri}(t)$ replaced by $h^e_{\text{tri}}(t)$ and with $s[n]$ replaced by the odd-index samples of $\theta_{\text{LTV}}[n]$ and $h_{tri}(t)$ replaced by $h^o_{\text{tri}}(t)$ ($h^e_{\text{tri}}(t)$ and $h^o_{\text{tri}}(t)$ are shown in Fig. 2.11(b)).

The analysis presented above proves that Fig. 2.11(a) represents a valid phase noise model of the PLL. The remainder of this appendix shows that the system of Fig. 2.5 is equivalent to that of Fig. 2.11(a).

The CTFT of the first summation in (42) can be evaluated as the product of the DTFT of $\theta_{\text{LTV}}[2n]$ and

$$H_T(\omega) = T_{\text{ref}} H(\omega) H_\mu(\omega) e^{-j\omega\Delta T}, \tag{43}$$

where $H(\omega)$ is given by (22) and

$$H_\mu(\omega) = \frac{1 + e^{j\omega T_{\text{ref}}\mu}\left(\left(\mu - 1\right)\cos\left(\omega T_{\text{ref}}\right) - \mu e^{-j\omega T_{\text{ref}}}\right)}{2(1-\mu^2)\sin^2\left(\omega T_{\text{ref}}/2\right)}. \tag{44}$$

Similarly, the CTFT of the second summation in (42) can be evaluated as the product of the DTFT of $\theta_{\text{LTV}}[2n+1]$ and

$$H_B(\omega) = T_{\text{ref}} H(\omega) H^*_\mu(\omega) e^{-j\omega(T_{\text{ref}} - \Delta T)}, \tag{45}$$

where $H^*_\mu(\omega)$ is the complex-conjugate of $H_\mu(\omega)$. Therefore, the FOH_e interpolator followed by the $-\Delta T$ time shift and the FOH_o interpolator followed by the $T_\text{ref} + \Delta T$ time shift in Fig. 2.11(a) represent multiplication in the frequency domain by $H_T(\omega)$ and $H_B(\omega)$, respectively.

The remainder of the proof utilizes a multi-rate system technique called block digital filtering [39]. Specifically, as illustrated in Fig. 2.13(a), any LTI transfer function, $H(z)$, with f_ref-rate input sequence, $x[n]$, can be parallelized and processed at a rate of $f_\text{ref}/2$ by a matrix transfer function,

$$\mathbf{H}(z^2) = \begin{bmatrix} H_0(z^2) & H_1(z^2) \\ z^{-2}H_1(z^2) & H_0(z^2) \end{bmatrix}, \tag{46}$$

called the blocked version of $H(z)$, where $H_0(z^2)$ and $H_1(z^2)$ are Type-I poly-phase components of $H(z)$ which satisfy

$$H(z) = H_0(z^2) + z^{-1}H_1(z^2)\,. \tag{47}$$

For example, the DCO transfer function, $K_\text{DCO}T_\text{ref}/(1–z^{-1})$, can be represented as in Fig. 2.13(a) with $\mathbf{H}(z^2)$ replaced by

$$\mathbf{H}_\mathbf{DCO}(z^2) = \frac{K_\text{DCO}T_\text{ref}}{1-z^{-2}}\begin{bmatrix} 1 & 1 \\ z^{-2} & 1 \end{bmatrix}. \tag{48}$$

Applying the block digital-filtering technique to the portion to the left of the $\theta_\text{LTV}[n]$ node in Fig. 2.11(a) results in the block diagram in Fig. 2.13(b), where $\mathbf{L}(z^2)$ and $\mathbf{H}_\mathbf{DCO}(z^2)$ are the blocked versions of $z^{-2}L(z)$ and $K_\text{DCO}T_\text{ref}/(1–z^{-1})$, respectively. As indicated in Fig. 2.13(b), the $1+\mu$ and $1–\mu$ multipliers can be implemented as $\mathbf{H}_\mu(z^2)$, where $\mathbf{H}_\mu(z^2)$ is specified in the figure, and the shaded cascade of up-sampling and down-sampling operations can be implemented as the identity matrix, $\mathbf{I}$. Two other cascades of up-sampling and down-sampling

operations, each of which can also be implemented as the identity matrix, occur in Fig. 2.13(b): one between $\mathbf{H}_{\mu}(z^2)$ and $\mathbf{H}_{\mathrm{DCO}}(z^2)$ and the other with the up-sampling operations to the right of $\mathbf{H}_{\mathrm{DCO}}(z^2)$ and the down-sampling operations to the left of $\mathbf{L}(z^2)$.

Applying these observations leads to the system shown in Fig. 2.14(a), where $\mathbf{P}(z) = [p_{ij}(z)]$ is the matrix product of $\mathbf{H}_{\mathrm{DCO}}(z)$, $\mathbf{H}_{\mu}(z)$, and $\mathbf{L}(z)$. Therefore,

$$\mathbf{P}(z) = p(z)\begin{bmatrix} 1+\mu & 1-\mu \\ z^{-1}(1+\mu) & 1-\mu \end{bmatrix}\begin{bmatrix} a(z) & b(z) \\ z^{-1}b(z) & a(z) \end{bmatrix}, \tag{49}$$

where

$$a(z) = \left[1+z^{-1}g(z)\right] + \lambda z^{-1}\left[1+g(z)\right], \tag{50}$$

$$b(z) = \lambda\left[1+z^{-1}g(z)\right] + \left[1+g(z)\right], \tag{51}$$

$$p(z) = K_I \frac{z^{-1}}{g(z)} \cdot \frac{K_{\mathrm{DCO}}T_{\mathrm{ref}}}{(1-z^{-1})^2} \cdot \left(\frac{1-\lambda}{1-\lambda^2 z^{-1}}\right), \tag{52}$$

and

$$g(z) = \frac{K_I}{K_P} \cdot \frac{1}{1+\left(\frac{K_I}{K_P}-1\right)z^{-1}}. \tag{53}$$

It follows from the expressions for $a(z)$, $b(z)$, and $p(z)$ above and tedious algebra that $T(z)$ in (26) can be written as

$$T(z) = p(z^2)(1+z^{-1})\left[a(z^2) + z^{-1}b(z^2)\right]. \tag{54}$$

As the up-sampling and down-sampling operations in the shaded box in Fig. 2.14(a) can be implemented as an identity matrix, Fig. 2.14(a) can be redrawn as shown in Fig. 2.14(b), where $\mathbf{A}(z) = [a_{ij}(z)]$ is given by

$$\mathbf{A}(z) = \frac{1}{D(z)} \begin{bmatrix} 1 + p_{22}(z) & -p_{12}(z) \\ -p_{21}(z) & 1 + p_{11}(z) \end{bmatrix} \mathbf{P}(z), \tag{55}$$

and

$$D(z) = \left[1 + p_{11}(z)\right]\left[1 + p_{22}(z)\right] - p_{12}(z)\, p_{21}(z)\,. \tag{56}$$

Substituting the elements of **P(***z***)** implied by (49) through (53) into (56) and applying (54) gives

$$D(z^2) = \left[1 + T(z)\right]\left[1 + T(-z)\right] - \mu^2 T(z) T(-z) \tag{57}$$

The results presented above show that the system of Fig. 2.11(a) is equivalent to that of Fig. 2.14(b).

The DTFTs of $x_T[n]$ and $x_B[n]$ in Fig. 2.14(b) are

$$X_T\left(e^{2j\omega T_{\text{ref}}}\right) = \frac{1}{2}\left(X\left(e^{j\omega T_{\text{ref}}}\right) + X\left(e^{j(\omega T_{\text{ref}} - \pi)}\right)\right), \tag{58}$$

and

$$X_B\left(e^{2j\omega T_{\text{ref}}}\right) = \frac{1}{2} e^{-j\omega T_{\text{ref}}}\left(X\left(e^{j\omega T_{\text{ref}}}\right) - X\left(e^{j(\omega T_{\text{ref}} - \pi)}\right)\right), \tag{59}$$

respectively [39]. It follows from (58) and (59) and the operations shown in Fig. 2.14(b) with expressions for the elements of **A(***z***)** given by (49) through (57) with $z = e^{j\omega T\text{ref}}$ that the CTFT of the sum of the outputs of $H_T(\omega)$ and $H_B(\omega)$ can be written as (23) with

$$C(\omega) = A_T\left(e^{2j\omega T_{\text{ref}}}\right) H_T(\omega) + A_B\left(e^{2j\omega T_{\text{ref}}}\right) H_B(\omega), \tag{60}$$

and

$$C^{\text{a}}(\omega) = A_T\left(e^{j(2\omega T_{\text{ref}} - \pi)}\right) H_T(\omega) + A_B\left(e^{j(2\omega T_{\text{ref}} - \pi)}\right) H_B(\omega), \tag{61}$$

where

$$A_T\left(e^{2j\omega T_{\text{ref}}}\right)=\frac{1}{2}G\left(e^{j\omega T_{\text{ref}}}\right)\left[E\left(e^{j\omega T_{\text{ref}}}\right)+F\left(e^{j\omega T_{\text{ref}}}\right)\right], \tag{62}$$

$$A_B\left(e^{2j\omega T_{\text{ref}}}\right)=\frac{e^{-j\omega T_{\text{ref}}}}{2}G\left(e^{j\omega T_{\text{ref}}}\right)\left[E\left(e^{j\omega T_{\text{ref}}}\right)-F\left(e^{j\omega T_{\text{ref}}}\right)\right], \tag{63}$$

$G\left(e^{j\omega T_{\text{ref}}}\right)$ is given by (26) with $z=e^{j\omega T_{\text{ref}}}$,

$$E\left(e^{j\omega T_{\text{ref}}}\right)=\frac{1+T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)-\mu^2 T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)}{1+T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)-\mu^2\dfrac{T\left(e^{j\omega T_{\text{ref}}}\right)T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)}{1+T\left(e^{j\omega T_{\text{ref}}}\right)}}, \tag{64}$$

and

$$F\left(e^{j\omega T_{\text{ref}}}\right)=\frac{(1-e^{-j\omega T_{\text{ref}}})\mu/(1+e^{-j\omega T_{\text{ref}}})}{1+T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)-\mu^2\dfrac{T\left(e^{j\omega T_{\text{ref}}}\right)T\left(e^{j(\omega T_{\text{ref}}-\pi)}\right)}{1+T\left(e^{j\omega T_{\text{ref}}}\right)}}. \tag{65}$$

Therefore, to prove that system of Fig. 2.5 is equivalent to those of Fig. 2.14(b) and Fig. 2.11(a), it is sufficient to derive expressions for $H_{\text{err}}(j\omega)$ and $H^a_{\text{err}}(j\omega)$ in Fig. 2.5 with which $C(\omega)$ and $C^a(\omega)$ given by (60) and (61) are equivalently given by (24) and (25). Substituting (44) into (43) and (45) and the results into (60) and (61) shows that (60) and (61) can be written as (24) and (25) with

$$H_{\text{err}}(j\omega)=E(e^{j\omega T_{\text{ref}}})\cdot\text{Re}\left\{H_\mu(\omega)e^{-j\omega T_{\text{ref}}\mu/2}\right\}+jF(e^{j\omega T_{\text{ref}}})\cdot\text{Im}\left\{H_\mu(\omega)e^{-j\omega T_{\text{ref}}\mu/2}\right\}, \tag{66}$$

and

$$H^a_{\text{err}}(j\omega)=F^a(e^{j\omega T_{\text{ref}}})\cdot\text{Re}\left\{H_\mu(\omega)e^{-j\omega T_{\text{ref}}\mu}\right\}+jE^a(e^{j\omega T_{\text{ref}}})\cdot\text{Im}\left\{H_\mu(\omega)e^{-j\omega T_{\text{ref}}\mu}\right\}. \tag{67}$$

ACKNOWLEDGEMENTS

This chapter, in full, is under review in the IEEE Transactions on Circuits and Systems I: Regular Papers. A. I. Eissa, E. Alvarez-Fontecilla, C. Weltin-Wu, I. Galton, 2024. The dissertation author is the primary investigator and author of this paper. Professor Ian Galton supervised the research which forms the basis for this paper.

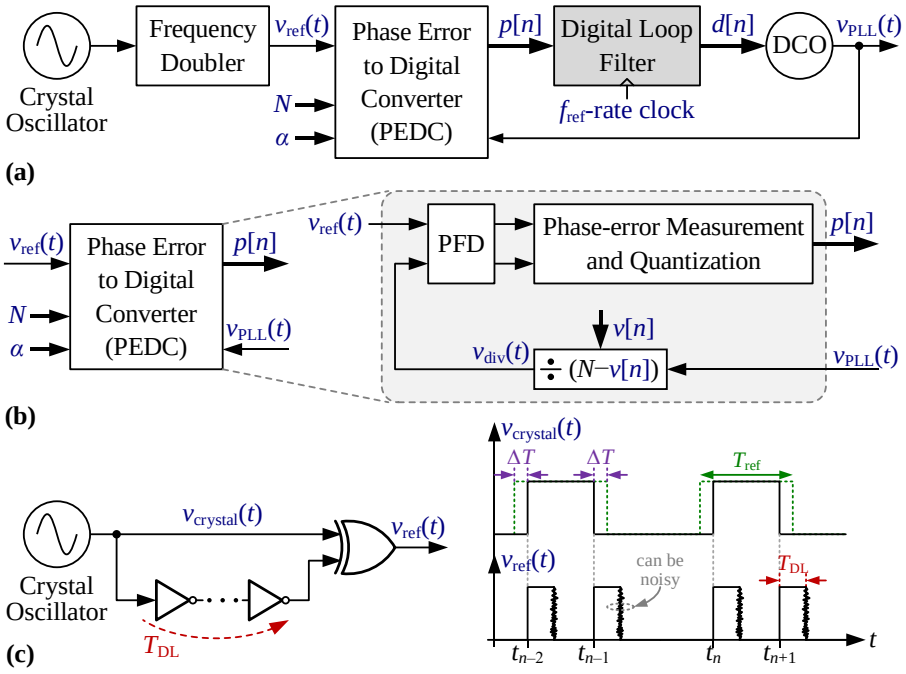


Figure 2.1. (a) Generic fractional-*N* digital PLL using a reference frequency-doubler, (b) general form of commonly used PEDCs that use MMD, and (c) XOR-based frequency-doubler and associated waveforms.

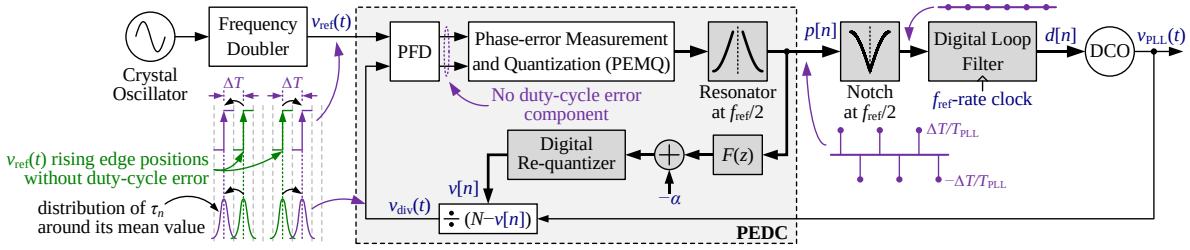


Figure 2.2. High-level block diagram of a generic fractional-*N* digital PLL with the proposed reference frequency-doubling scheme.

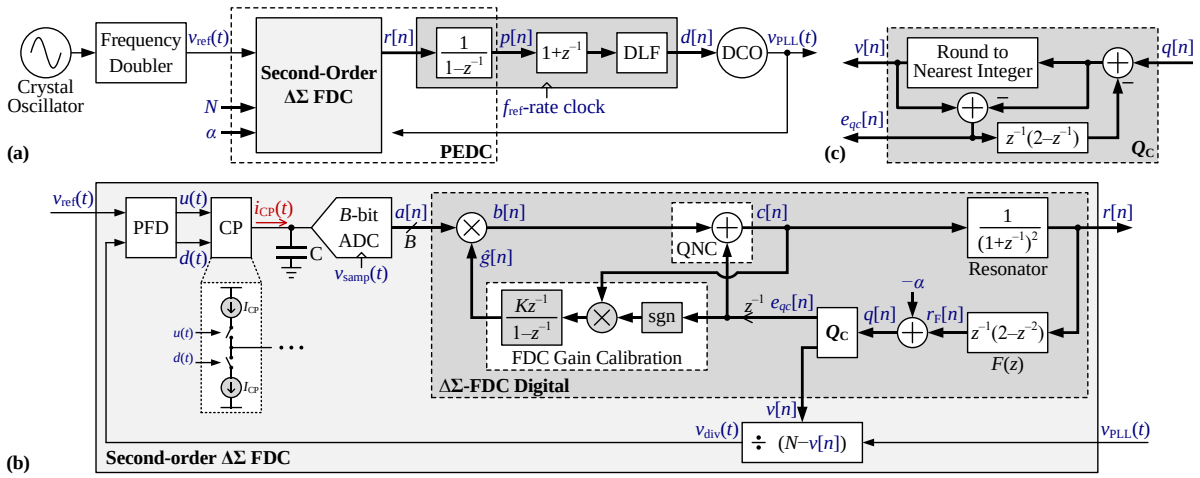


Figure 2.3. Block diagram of the proposed FDC-PLL: (a) PLL top-level block diagram, (b) second-order ΔΣ-FDC block diagram, and (c) coarse-quantizer, Q_C, implementation details.

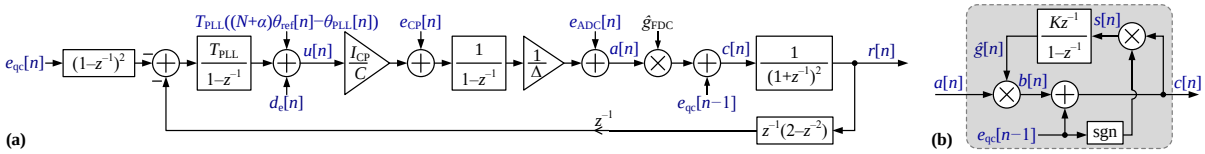


Figure 2.4. (a) Linearized model of the proposed ΔΣ-FDC where $\hat{g}[n] = \hat{g}_{FDC}$ is approximated as constant, and (b) FDC gain calibration behavioral model.

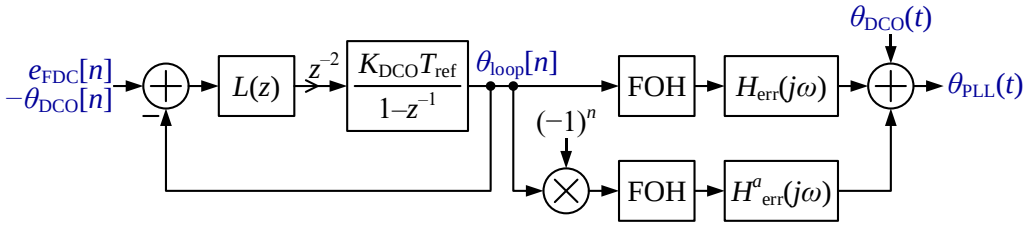


Figure 2.5. PLL LTV phase noise model.

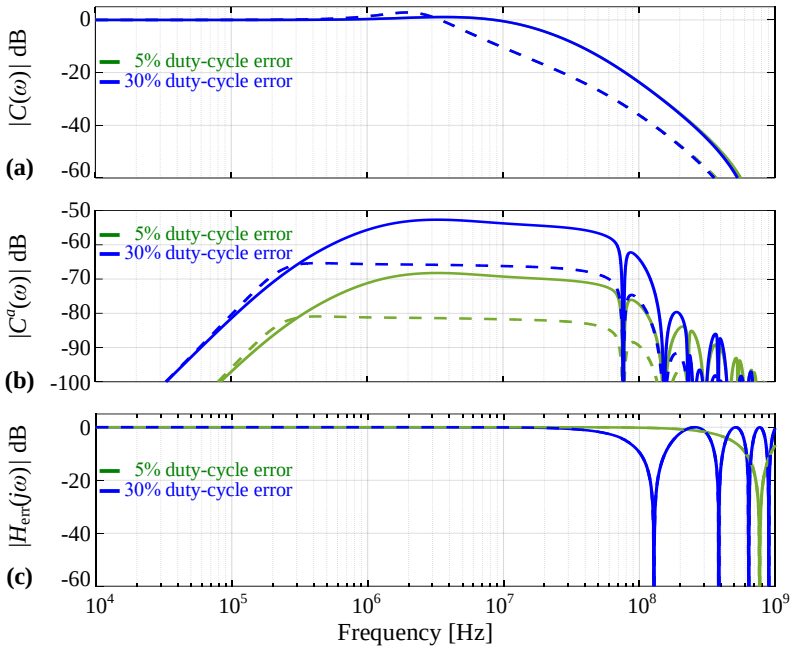


Figure 2.6. Magnitude responses of (a) C(ω), (b) $C^q(\omega)$, and (c) $H_{err}(j\omega)$. Dashed (solid) lines correspond to a PLL bandwidth of 280 kHz (1.3 MHz). The 5% and 30% duty-cycle error curves coincide in (a).

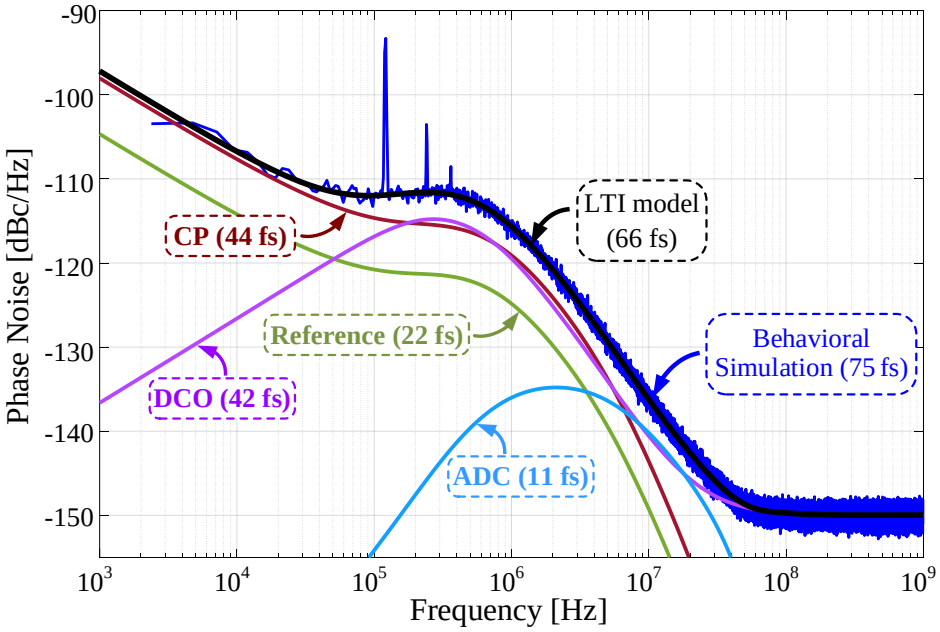


Figure 2.7. PLL phase noise power spectra.

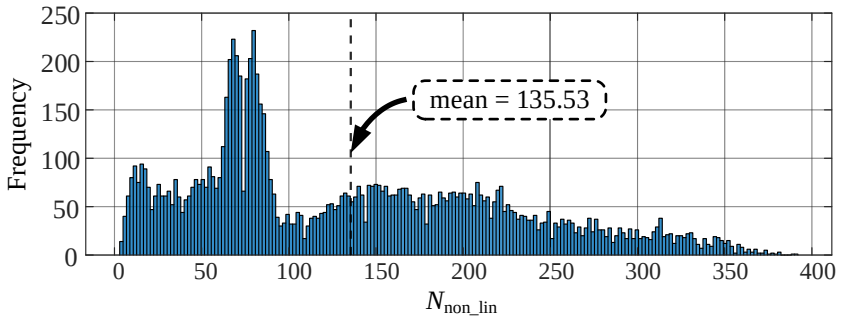


Figure 2.8. Histogram of $N_{non\text{-}lin}$ for 10000 PLL runs with random initial conditions.

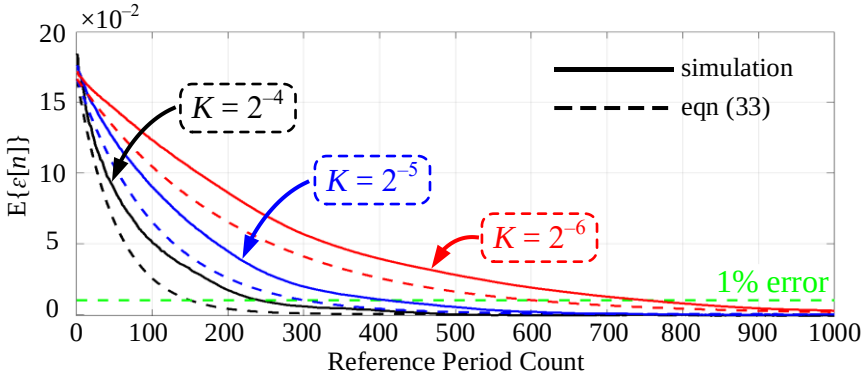


Figure 2.9. FDC gain calibration error sequence, $E\{\varepsilon[n]\}$.

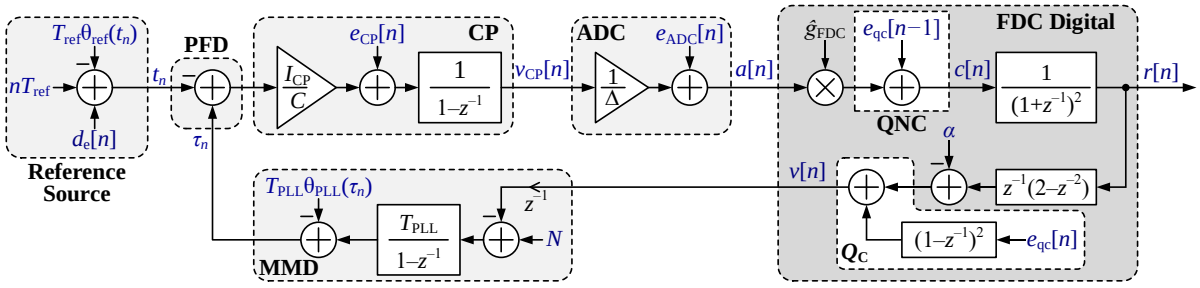


Figure 2.10. Linearized behavioral model of the proposed ΔΣ-FDC where $\hat{g}[n] = \hat{g}_{FDC}$ is approximated as constant.

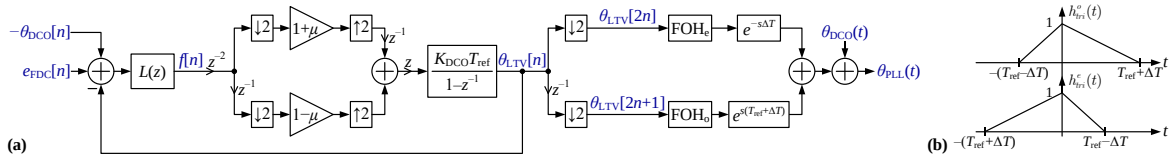


Figure 2.11. (a) PLL's LTV model and (b) the FOH_e and FOH_o interpolation functions, $h^e_{tri}(t)$ and $h^o_{tri}(t)$.

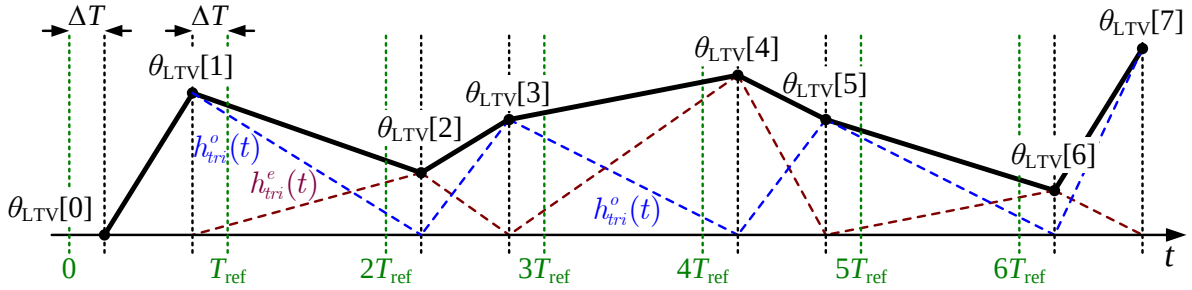


Figure 2.12. Non-uniform linear interpolation between the $\theta_{LTV}[n]$ samples.

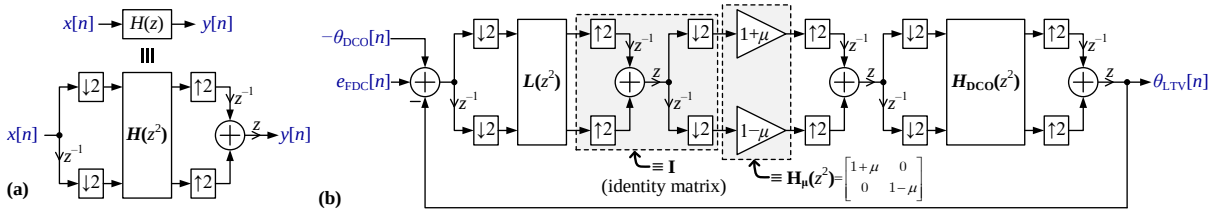


Figure 2.13. (a) Block digital filtering technique, and (b) block digital filtering applied to the PLL's LTV phase noise model of Fig. 2.13(a).

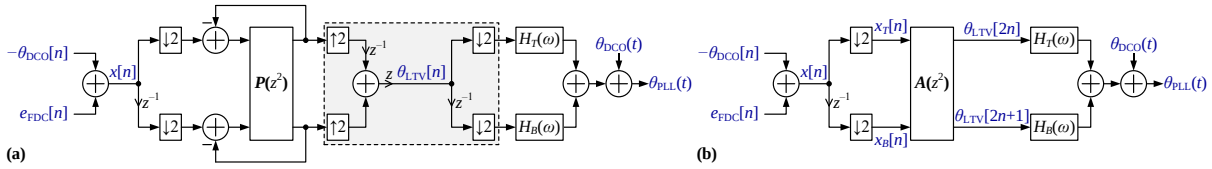


Figure 2.14. (a) PLL LTV phase noise model after applying the block digital filtering technique, and (b) simplified representation.

TABLES

Table 2.1. Contribution of different noise sources to the PLL's output.

Noise Source	Contribution to the PSD of $\theta_{\mathrm{PLL}}(t)$ in dBc/Hz
Reference Source	$4\pi^2 \left\lvert (N+\alpha) G(e^{j\omega T_{\mathrm{ref}}}) H(\omega) \right\rvert^2 S_{\mathrm{ref}}(\omega)$
CP	$4\pi^2 T_{\mathrm{ref}} \left\lvert (\hat{g}_{\mathrm{FDC}} / \Delta) G(e^{j\omega T_{\mathrm{ref}}}) H(\omega) \right\rvert^2 S_{\mathrm{CP}}(e^{j\omega T_{\mathrm{ref}}})$
ADC	$4\pi^2 (T_{\mathrm{ref}} / 3) \left\lvert 2^{-F} \hat{g}_{\mathrm{FDC}} \sin(\omega T_{\mathrm{ref}} / 2) G(e^{j\omega T_{\mathrm{ref}}}) H(\omega) \right\rvert^2$
DCO	$4\pi^2 \left\lvert 1 - G(e^{j\omega T_{\mathrm{ref}}}) H(\omega) \right\rvert^2 S_{\mathrm{DCO}}(\omega)$

$S_{\mathrm{ref}}(\omega)$ and $S_{\mathrm{DCO}}(\omega)$ are the two-sided phase-noise PSDs of the reference and DCO, respectively, in cycles squared per Hz, and $S_{\mathrm{CP}}(e^{j\omega T_{\mathrm{ref}}})$ is the two-sided discrete-time PSD of $e_{\mathrm{CP}}[n]$ in dBV/Hz.

Table 2.2. PLL design parameters used for the behavioral simulation

Design Parameters		Value	
Crystal Oscillator	Frequency, f_{crystal}	76.8 MHz	
	Phase noise(1)	−160 and −163 dBc/Hz	
Frequency Doubler	Added phase noise	Negligible	
	T_{DL}	3.25 ns	
Charge Pump	Total current	2 mA	
	Nominal capacitance	1 pF	
	Noise(2)	−146 and −156 dBV/Hz	
ADC	Number of bits (B, F)	7, 5	
	Coarse step-size, Δ	100 mV	
DCO	DCO gain, K_{DCO}	150 kHz	
	Phase noise(3)	−122, −120, and −150 dBc/Hz	
DLF	Proportional gain, K_P	5	20
	Integral gain, K_I	0.0390625	0.15625
	IIR pole, λ	0.75	0.75
FDC Digital	FDC gain calibration, K	0.015625	
PLL Settings	Integer multiplier, N	65	
	Fractional multiplier, α	0.0007848739624	
	Output frequency, f_{PLL}	10 GHz	
	Loop bandwidth	280 kHz	1.3 MHz
	RMS Jitter	113 fs	75 fs

[1] $1/f$ and white phase noise components at 10 kHz offset.
[2] $1/f$ and white discrete-time PSD of $e_{\mathrm{CP}}[n]$ at 10 kHz offset.
[3] $1/f^3$, $1/f^2$ and white phase noise components at 1 MHz offset.

Table 2.3. Duty-cycle error convergence time comparison

Reference	Initial Duty-Cycle error	Convergence time in reference cycles
[11]	6.00%	2000
[12]	7.00%	3800
[16]	0.25%	6000
[18]	0.50%	2500
This work	−10% to 10%	412†

† Worst-case convergence time across 10000 PLL runs with random duty-cycle error and PLL initial conditions.

REFERENCES

1. W. Wu, "Low-Jitter Frequency Generation Techniques for 5G Communication: A tutorial," in *IEEE Solid-State Circuits Magazine*, vol. 13, no. 4, pp. 44-63, Fall 2021.

2. M. Oveisi and P. Heydari, "A Study of BER and EVM Degradation in Digital Modulation Schemes Due to PLL Jitter and Communication-Link Noise," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 8, pp. 3402-3415, Aug. 2022.

3. S. Levantino, "Recent Advances in High-Performance Frequency Synthesizer Design," *2022 IEEE Custom Integrated Circuits Conference (CICC)*, Newport Beach, CA, USA, 2022, pp. 1-7.

4. M. Shinagawa, Y. Akazawa and T. Wakimoto, "Jitter analysis of high-speed sampling systems," in *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 220-224, Feb. 1990.

5. B. Razavi, "Jitter-Power Trade-Offs in PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 4, pp. 1381-1387, April 2021.

6. B. Razavi, "Lower Bounds on Power Consumption of Clock Generators for ADCs," *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020, pp. 1-5.

7. P. M. Aziz, H. V. Sorensen and J. vn der Spiegel, "An overview of sigma-delta converters," in *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 61-84, Jan. 1996.

8. SiTime, "*Ultra-Performance Oscillator*," SiT8209 datasheet, Jan. 2017.

9. Crystek, "*CVHD-950 VCXO Ultra-Low Phase Noise Oscillators*," CVHD-950 VCXO datasheet, Aug. 2019.

10. Crystek, "*CVHD-957 Ultra-Low Phase Noise VCXO with Standby Mode*," CVHD-959 VCXO datasheet, Aug. 2019.

11. W. Wu *et al*., "A 28-nm 75-fsrms Analog Fractional-*N* Sampling PLL With a Highly Linear DTC Incorporating Background DTC Gain Calibration and Reference Clock Duty Cycle Correction," in *IEEE Journal of Solid-State Circuits*, vol. 54, no. 5, pp. 1254-1265, May 2019.

12. 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," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 12, pp. 3756-3767, Dec. 2021.

13. S. Abdollahi-Alibeik *et al.*, "A 65nm dual-band 3-stream 802.11n MIMO WLAN SoC," *2011 IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2011, pp. 170-172.

14. Y. -L. Hsueh *et al.*, "28.2 A 0.29mm2 frequency synthesizer in 40nm CMOS with 0.19psrms jitter and <-100dBc reference spur for 802.11ac," *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, San Francisco, CA, USA, 2014, pp. 472-473.

15. T. Siriburanon *et al.*, "A 28-GHz fractional-N frequency synthesizer with reference and frequency doublers for 5G cellular," *ESSCIRC Conference 2015 - 41st European Solid-State Circuits Conference (ESSCIRC)*, Graz, Austria, 2015, pp. 76-79.

16. H. Kim, Y. Kim, T. Kim, H. -J. Ko and S. Cho, "A 2.4-GHz 1.5-mW Digital Multiplying Delay-Locked Loop Using Pulsewidth Comparator and Double Injection Technique," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 11, pp. 2934-2946, Nov. 2017.

17. H. Yoon *et al.*, "A −31dBc integrated-phase-noise 29GHz fractional-N frequency synthesizer supporting multiple frequency bands for backward-compatible 5G using a frequency doubler and injection-locked frequency multipliers," *2018 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2018, pp. 366-368.

18. A. Elkholy, D. Coombs, R. K. Nandwana, A. Elmallah and P. K. Hanumolu, "A 2.5–5.75-GHz Ring-Based Injection-Locked Clock Multiplier With Background-Calibrated Reference Frequency Doubler," in *IEEE Journal of Solid-State Circuits*, vol. 54, no. 7, pp. 2049-2058, July 2019.

19. C. -W. Yao, L. Lin, B. Nissim, H. Arora and T. Cho, "A low spur fractional-N digital PLL for 802.11 a/b/g/n/ac with 0.19 psrms jitter," *2011 Symposium on VLSI Circuits - Digest of Technical Papers*, Kyoto, Japan, 2011, pp. 110-111.

20. X. Gao *et al.*, "9.4 A 28nm CMOS digital fractional-N PLL with −245.5dB FOM and a frequency tripler for 802.11abgn/ac radio," *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, USA, 2015, pp. 1-3.

21. C. Venerus and I. Galton, "Delta-Sigma FDC Based Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 5, pp. 1274-1285, May 2013.

22. C. Venerus and I. Galton, "Quantization Noise Cancellation for FDC-Based Fractional-*N* PLLs," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 12, pp. 1119-1123, Dec. 2015.

23. C. -M. Hsu, M. Z. Straayer and M. H. Perrott, "A Low-Noise Wide-BW 3.6-GHz Digital ΔΣ Fractional-N Frequency Synthesizer With a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2776-2786, Dec. 2008.

24. M. Zanuso, S. Levantino, C. Samori and A. L. Lacaita, "A Wideband 3.6 GHz Digital ΔΣ Fractional-N PLL With Phase Interpolation Divider and Digital Spur Cancellation," in *IEEE Journal of Solid-State Circuits*, vol. 46, no. 3, pp. 627-638, March 2011.

25. C. -W. Yao *et al*., "A 14-nm 0.14-psrms Fractional-N Digital PLL With a 0.2-ps Resolution ADC-Assisted Coarse/Fine-Conversion Chopping TDC and TDC Nonlinearity Calibration," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 12, pp. 3446-3457, Dec. 2017.

26. L. Bertulessi, D. Cherniak, M. Mercandelli, C. Samori, A. L. Lacaita and S. Levantino, "Novel Feed-Forward Technique for Digital Bang-Bang PLL to Achieve Fast Lock and Low Phase Noise," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 5, pp. 1858-1870, May 2022.

27. C. Weltin-Wu, E. Familier and I. Galton, "A Linearized Model for the Design of Fractional-N Digital PLLs Based on Dual-Mode Ring Oscillator FDCs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 8, pp. 2013-2023, Aug. 2015.

28. M. Talegaonkar *et al*., "A 5GHz Digital Fractional- N PLL Using a 1-bit Delta–Sigma Frequency-to-Digital Converter in 65 nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 9, pp. 2306-2320, Sept. 2017.

29. E. Helal, E. Alvarez-Fontecilla, A. I. Eissa and I. Galton, "A Time Amplifier Assisted Frequency-to-Digital Converter Based Digital Fractional-N PLL," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2711-2723, Sept. 2021.

30. E. Helal, A. I. Eissa and I. Galton, "DTC Linearization via Mismatch-Noise Cancellation for Digital Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 12, pp. 4993-5006, Dec. 2022.

31. I. Galton and C. Weltin-Wu, "Understanding Phase Error and Jitter: Definitions, Implications, Simulations, and Measurement," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 1-19, Jan. 2019.

32. B. Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge: Cambridge University Press, 2020.

33. N. D. Dalt, "Markov Chains-Based Derivation of the Phase Detector Gain in Bang-Bang PLLs," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 11, pp. 1195-1199, Nov. 2006.

34. J. Proakis and D. Manolakis, *Digital Signal Processing*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 1996.

35. E. Alvarez-Fontecilla, A. I. Eissa, E. Helal, C. Weltin-Wu and I. Galton, "Delta-Sigma FDC Enhancements for FDC-Based Digital Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 965-974, March 2021.

36. K. J. Wang, A. Swaminathan and I. Galton, "Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4 GHz Fractional-N PLL," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2787-2797, Dec. 2008.

37. A. Yu, D. Bankman, K. Zheng and B. Murmann, "Understanding Metastability in SAR ADCs: Part II: Asynchronous," in *IEEE Solid-State Circuits Magazine*, vol. 11, no. 3, pp. 16-32, Summer 2019.

38. R. B. Staszewski and P. T. Balsara, "Phase-domain all-digital phase-locked loop," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 3, pp. 159-163, March 2005.

39. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.

CHAPTER 3

A CHARGE PUMP NONLINEARITY MECHANISM AND ITS MITIGATION IN FRACTIONAL-*N* PLLS

Abstract— Spurious tones generated in fractional-*N* phase-locked loops (PLLs) can limit a PLL's achievable jitter. Spurious tones can be generated by subjecting DC-free quantization error and its running sum to nonlinear distortion such as that from a charge pump (CP) circuit. In this paper, a CP parasitic-capacitance-induced nonlinearity mechanism is described along with a scheme to mitigate it. Behavioral simulations with nonideal circuit parameters show that the proposed technique attenuates the spurious tones by more than 10 dB in a 10-GHz delta-sigma frequency-to-digital converter (ΔΣ-FDC) PLL.

I. INTRODUCTION

High data rates in wireless and wireline communication systems require phase-locked loops (PLLs) with a sub-100-fs RMS jitter [1], [2], [3], [4], [5], [6]. The power spectral density (PSD) of a fractional-*N* PLL's phase error contains a random and a deterministic component, typically referred to as *phase noise* and *spurious tones*, respectively [7]. Minimizing both components is critical to achieving stringent PLL jitter requirements.

A fundamental source of spurious tones comes from a fractional-*N* PLL's DC-free quantization error, which is generated and subsequently lowpass filtered as part of the fractional-*N* PLL normal operation [8], [9], [10], [11], [12]. Subjecting the quantization error and its running sum to nonlinear distortion introduces spurious tones to the PLL's output even

when the quantization error itself is free of spurious tones [10], [11]. Moreover, nonlinear distortion causes the quantization error to fold within the PLL's bandwidth, further increasing the PLL's total RMS jitter, σ_{JT} [13], [14].

A dominant source of nonlinearity in a wide class of fractional-*N* PLLs is the charge pump (CP) [13], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. In this paper, a CP parasitic-capacitance-induced nonlinearity mechanism is described, and a linearization scheme to mitigate it is presented. The scheme is presented in the context of a 10 GHz delta-sigma frequency-to-digital converter (ΔΣ-FDC) PLL. Behavioral simulations with nonideal circuit parameters extracted from transistor-level circuit simulations show that the proposed scheme reduces the spurious tones' power by more than 10 dB, achieving a worst-case in-band spur level below −54 dBc and σ_{JT} below 80 fs.

II. CP NONLINEARITY MECHANISM AND ITS MITIGATION

Fig. 3.1(a) shows a top-level block diagram of a fractional-*N* PLL that fits a wide class of analog and digital PLLs [8], [13], [15], [16], [17], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. The PLL generates a periodic output signal, $v_{PLL}(t)$, with frequency $f_{PLL} = (N + \alpha)f_{ref}$, where N is a positive integer, $-½ \leq \alpha \leq ½$, and f_{ref} is the frequency of the PLL's reference oscillator signal, $v_{ref}(t)$.

The nth and $(n+1)$th rising edges of the multi-modulus divider (MMD) output, $v_{div}(t)$, are separated by $N - v[n]$ PLL cycles, where $v[n]$ is an integer-valued sequence generated within the PLL. During the nth reference period, the phase-frequency detector (PFD) compares the times of the rising edges of $v_{ref}(t)$ and $v_{div}(t)$, t_n and τ_n respectively, and generates $u(t)$ and $d(t)$.

Ideally, as shown in Fig. 3.1(b), the PFD causes the CP to output a current pulse, $i_{CP}(t)$, with a width of $|\tau_n - t_n|$ and nominal amplitude of I_{CP} when $t_n < \tau_n$ and $-I_{CP}$ when $\tau_n < t_n$. In conventional analog PLLs, $i_{CP}(t)$ is lowpass filtered by an analog loop filter, and the resulting waveform drives the controlled oscillator [8], [16], and [22].[4] In digital PLLs, such as in [23], [27], and [29], $i_{CP}(t)$ accumulates charge over a capacitor whose voltage is then sampled by an analog-to-digital converter (ADC). The ADC's output is used to generate a measure of the PLL's phase error that is lowpass filtered before controlling the frequency of the controlled oscillator [13].

The MMD input sequence $v[n]$ can be generated by a digital re-quantizer, as in [15], [17], [20], and [30], or by filtering the PLL's quantized phase error sequence, as in [23], [31], [32], [33], and [34]. In each case, $v[n]$ contains a zero-mean, highpass-shaped coarse quantization error component, $s[n]$, and the PLL settles such that $v[n]$ has a mean of $-\alpha$, so the PLL's mean output frequency is $(N + \alpha)f_{ref}$. When $s[n]$, or its running sum, $t[n]$, are subjected to nonlinear distortion from the CP, spurious tones at integer multiples of αf_{ref} (known as *fractional spurs*) are generated, degrading the PLL's jitter [9], [10], [11], [12], [18].

A. *CP Nonlinearity Mechanism*

Inevitable device mismatches and channel-length modulation effects cause the CP currents I_P and I_N in Fig. 3.1(b) to have different values, which introduces nonlinearity as this causes the magnitude of the CP current pulses to depend on $|\tau_n - t_n|$. Fig. 3.2(a) shows an offset-current linearization technique commonly used to address this issue. As shown in the figure, a current source I_{OC}, nominally equal to $I_N = I_{CP}$, is added to the CP, which is controlled by a

[4] The controlled oscillator is a voltage-controlled oscillator in the case of an analog PLL and a digitally-controlled oscillator in the case of a digital PLL.

pulse $u_{OC}(t)$ of fixed width T_{OC} [13], [15], [17], [25]. As the I_{OC} current source dumps a fixed amount of charge each reference period, the PLL counteracts it by locking to a state such that $\tau_n < t_n$ for all n and $d(t)$ has a mean equal to T_{OC}. Consequently, the I_P current source plays no role in the PLL operation after locking, and the I_P/I_N mismatch is avoided [13], [15], [17], [26].

Fig. 3.2(a) also illustrates the operation of a CP with the offset-current linearization technique. The figure shows the four phases of operation (φ_1 through φ_4) during which either of $u_{OC}(t)$ or $d(t)$ is high. As shown in Fig. 3.2(b), the CP is assumed to drive a capacitor C, C_{OC} and C_N represent the parasitic capacitances associated with I_{OC} and I_N, respectively, and ideally $I_{OC} = I_N = I_{CP}$. During φ_1 and φ_3, the voltage across C, C_N, and C_{OC} does not change. During φ_2 and φ_4, I_{CP} is sourced to $C + C_{OC}$ or sunk from $C + C_N$ depending on whether the pulse width of $d(t)$ is smaller or larger than T_{OC}, respectively. This causes a discontinuity in the CP's output voltage slew rate when the pulse width of $d(t)$ is around T_{OC}, which ultimately introduces nonlinearity.[5]

As the CP current is integrated by the capacitors, the settled value of the CP voltage across the capacitor during the nth reference period, $v_{CP}[n]$, can be expressed as

$$v_{CP}[n] = v_{CP}[n-1] + \Delta v_{CP}[n], \tag{68}$$

where

$$\Delta v_{CP}[n] = \begin{cases} \left(\dfrac{I_{OC} - I_N}{C + C_N + C_{OC}} - \dfrac{I_{OC}}{C + C_{OC}} \right) d[n] + \dfrac{I_{OC}}{C + C_{OC}} T_{OC} & \text{for } d[n] < T_{OC}, \text{ and} \\ -\dfrac{I_N}{C + C_N} d[n] + \left(\dfrac{I_{OC} - I_N}{C + C_N + C_{OC}} + \dfrac{I_N}{C + C_N} \right) T_{OC} & \text{for } d[n] > T_{OC} \end{cases}, \tag{69}$$

[5] Although presented in the context of a CP with offset-current, this nonlinearity mechanism affects the conventional CP scheme in Fig. 3.1(b) as well.

and $d[n]$ is the pulse width of $d(t)$ during the nth reference cycle.

For example, a PLL design with $I_{OC} = I_N$, $C + C_{OC} = 1.01$ pF, and $C + C_N = 0.99$ pF, the $C_{OC} - C_N = 20$ fF difference results in a 2% mismatch in the CP's output voltage slew rate around $d[n] = T_{OC}$. Behavioral simulations for the 10 GHz PLL architecture described in Section III show that such mismatch level results in a –43 dBc fractional-N spur, contributing 160 fs to the PLL's jitter. The parasitic-capacitance-induced nonlinearity becomes more prominent in PLLs with relatively small C values, as in analog PLLs with large bandwidths or in digital PLLs [17], [19], [25], [27], [29]. This issue is exacerbated in CPs running from low supply voltages or used in low-jitter PLLs. Both cases require relatively large transistors to reduce their required voltage headroom and sufficiently suppress their flicker noise, respectively [13]. As I_{OC} and I_N are typically implemented using PMOS and NMOS devices, respectively, that have different mobilities, the design trade-offs described above result in larger $C_{OC} - C_N$ values [35], resulting in higher spur and jitter levels.

B. *Proposed CP Linearization Scheme*

Fig. 3.3(a) shows the proposed scheme to mitigate the nonlinearity mechanism described above. The idea is to nominally set $I_{OC} = 0.5I_N = 0.5I_{CP}$ and double the pulse width of $u_{OC}(t)$ to $2T_{OC}$. This results in the same amount of fixed charge dumped every reference cycle as with the conventional scheme; hence, the mean of $d[n]$ is still T_{OC}. However, as shown in Fig. 3.3(b), the CP equivalent schematic is identical during φ_2 and φ_4, provided that $2T_{OC}$ is larger than the maximum value of $d[n]$. This eliminates the slew rate discontinuity associated with the conventional offset current linearization scheme. Applying the same reasoning above to the CP operation in Fig. 3.3 yields:

$$\Delta v_{\mathrm{CP}}[n] = \left(\frac{I_{\mathrm{OC}} - I_{\mathrm{N}}}{C + C_{\mathrm{N}} + C_{\mathrm{OC}}} - \frac{I_{\mathrm{OC}}}{C + C_{\mathrm{OC}}} \right) d[n] + \frac{2I_{\mathrm{OC}}}{C + C_{\mathrm{OC}}} T_{\mathrm{OC}}, \tag{70}$$

which is linear across the entire $d[n]$ range. Fig. 3.3(c) summarizes the differences between the conventional and proposed offset-current linearization schemes.

Compared to the conventional offset-current scheme, the PSD of the I_{OC} white current-noise contribution to the PLL's phase noise, $S_{\mathrm{OC,W}}(f)$, does not change, whereas that of the flicker noise, $S_{\mathrm{OC,F}}(f)$, increases by 3 dB. The PSDs $S_{\mathrm{OC,W}}(f)$ and $S_{\mathrm{OC,F}}(f)$ are proportional to $T_{\mathrm{ON}}S_{\mathrm{I,W}}(f)$ and $T^2_{\mathrm{ON}}S_{\mathrm{I,F}}(f)$, respectively, where T_{ON} is the average duration during which I_{OC} dumps current into the PLL, and $S_{\mathrm{I,W}}(f)$ and $S_{\mathrm{I,F}}(f)$ are the I_{OC} white and flicker current noise PSDs, respectively [13]. As $S_{\mathrm{I,W}}(f)$ is proportional to I_{OC}, the product $T_{\mathrm{ON}}S_{\mathrm{I,W}}(f)$ does not change with the proposed scheme. For the flicker noise, $S_{\mathrm{I,F}}(f)$ is proportional to I^2_{OC}/WL, where W and L are the width and length, respectively, of the transistor generating I_{OC}. Therefore, halving I_{OC} by halving its transistor's width reduces $S_{\mathrm{I,F}}(f)$ by 3 dB. As T^2_{OC} is quadrupled, $T^2_{\mathrm{ON}}S_{\mathrm{I,F}}(f)$ increases by 3 dB. Usually the white noise component is more dominant, so the increase in $S_{\mathrm{OC,F}}(f)$ does not significantly affect the PLL's jitter. In cases where it does, $S_{\mathrm{OC,F}}(f)$ can be reduced by increasing both W and L by the same factor (e.g., 1.414 to reduce $S_{\mathrm{I,F}}(f)$ by 3 dB).

III. IMPLEMENTATION DETAILS

The ΔΣ-FDC based PLL in [36] is used to demonstrate the nonlinearity mechanism and the proposed mitigation scheme.

A. *PLL Architecture Overview*

Fig. 3.4(a) shows the PLL's architecture. It comprises a second-order ΔΣ-FDC that generates a quantized measure of the PLL's frequency error, $r[n]$. The digital loop controller (DLC) accumulates $r[n]$ to generate a phase error sequence that is lowpass filtered before controlling the frequency of the digitally controlled oscillator (DCO). Fig. 3.4(b) shows a simplified block diagram of the ΔΣ-FDC. As the details of the PLL and ΔΣ-FDC are explained in [36], only the implementation details relevant to the CP are presented here.

The sequence $v[n]$ has a highpass shaped quantization error component, $s[n]$, generated from a second-order digital delta-sigma modulator, such that

$$s[n] = e_{qc}[n] - 2e_{qc}[n-1] + e_{qc}[n-2], \tag{71}$$

where $e_{qc}[n]$ is the quantization error sample introduced by the quantizer in the digital delta-sigma modulator. This MMD control scenario is equivalent to that in a conventional analog fractional-N PLL [8]. The CP in Fig. 3.4(b) uses the proposed offset current scheme presented in Section II-B. After the PLL locks, the $u(t)$ signal is always low, and $d[n]$ has a component proportional to the running sum of $s[n]$. The CP current is integrated by the capacitor, C, and the B-bit ADC samples the CP's output voltage at the rising edges of $v_{samp}(t)$, which is a delayed version of $v_{ref}(t)$.

B. *CP Design*

Fig. 3.5 shows the CP circuit topology, along with the devices' sizing, implemented in the Global Foundries 22-nm CMOS FDSOI process. A current steering topology is chosen to ensure fast current settling, mitigating the nonlinearity associated with incomplete current

settling, and allowing for the use of short $u_{OC}(t)$ pulse widths to reduce the CP noise [13]. The offset and CP currents, I_{OC} and I_N, are implemented as PMOS and NMOS current sources, respectively, using the low-voltage cascode topology (bias details not shown) to boost their output resistance [35]. The choice of sourcing a fixed amount of charge each reference cycle rather than sinking it maintains a signal independent CP supply activity, reducing the potential of fractional spur generation through supply coupling. The sizing and bias levels of transistors $M_{1,2,7,8}$ was determined based on noise and voltage headroom requirements.

The currents I_{OC} and I_N are steered to the CP output whenever $u_{OC}(t)$ and $d(t)$ are high, respectively. Otherwise, they are steered to a dummy node, $V_d(t)$, where a dummy PMOS current source equal to I_{OC} is connected so that I_N is balanced when both $u_{OC}(t)$ and $d(t)$ are low. A source follower stage is added such that $V_d(t)$ settles to the same level (around mid-supply voltage) each cycle before the next $u_{OC}(t)$ and $d(t)$ pulses arrive. This ensures that charge sharing between C and C_d, that takes place momentarily during the $d(t)$ transitions, does not introduce nonlinear distortion. The steering switches, $M_{3,4,5,6}$, use minimum channel-length to reduce their ON resistance. Parametric sweeps were used to optimize the switches' width value as larger widths reduce the switches' ON resistance but degrade linearity due to charge injection and clock feedthrough.

C. *Simulation Results*

The PLL design example presented in this section has f_{ref} = 153.6 MHz, f_{PLL} = 10 GHz, and achieves σ_{JT} less than 80 fs. Table I summarizes the relevant design parameters and noise contributions. The noise contributions and other nonideal circuit behavior, such as CP nonlinearity, were determined via Cadence Spectre simulations of the respective circuits

implemented in the Global Foundries 22-nm CMOS FDSOI process. As in [36], parameters that describe the nonideal circuit behavior were extracted from the transistor-level simulations and back-annotated into a custom, C-language, event-driven, bit-exact, behavioral simulator.

To capture the CP's nonlinear behavior, the CP's transistor-level simulated output voltages versus the expected range of PFD output pulse-widths, in increments of 5 ps, were back-annotated into a look-up table (LUT), which the behavioral simulator uses to calculate each CP output voltage via piecewise linear interpolation between adjacent LUT entries. The transistor-level simulation testbench included realistic models for the supply network (including the supply source impedance, routing traces, bond-wires, and decoupling capacitors) to capture the effect of the PFD transitions on the power supply shared with the CP. The offset-current pulse width, $2T_{OC}$, was set to 300 ps, which simulations predicted is sufficient for supply ripples to not significantly degrade the CP linearity.

Fig. 3.6 shows the simulated PLL's phase noise profile, with and without the proposed scheme along, with individual noise contributions based on the parameters in Table I and the linearized model derived in [36]. The integrated total jitter, σ_{JT}, is 75.8 fs and 188.3 fs with and without the proposed scheme, respectively, where the integration band extends from 500 Hz to 200 MHz. With the proposed offset-current scheme, only the first two fractional spurs are significant, and their power levels are –55.2 dBc and –69.8 dBc, contributing 40 fs and 7 fs to σ_{JT}, respectively. With the conventional offset-current scheme, the first four fractional spurs are significant, and their powers are –42.6 dBc, –54.6 dBc, –58.9, –63.6 dBc, contributing 167.11 fs, 42 fs, 25.6 fs, and 14.9 fs to σ_{JT}, respectively. In addition, quantization error folding, in the conventional offset-current case, adds around 28 fs to σ_{JT}. This folding effect is evident in Fig.

3.6 as the phase noise profile with the conventional offset-current scheme is slightly higher than the profile with the proposed one.

To comprehensively evaluate the fractional spur performance of the PLL, the fractional spur frequency was swept between 500 Hz and 10 MHz, and for each run, σ_{JT} and the worst-case fractional spur level were recorded. Fig. 3.7 shows the results of this sweep for the conventional and proposed offset-current schemes. As shown, enabling the proposed scheme reduces the worst-case fractional spur power by more than 10 dB. For fractional spur frequencies less than 1 MHz, the proposed scheme resulted in a jitter reduction between 83 fs and 115 fs. Beyond 1 MHz, the fractional spurs are attenuated by the PLL's loop filter and the difference in σ_{JT} between the conventional and proposed schemes gets progressively smaller as the random jitter component, common to both schemes, becomes more dominant.

IV. CONCLUSION

The parasitic capacitances associated with the CP current sources cause a discontinuity in the CP's output voltage slew rate even when using the conventional offset current linearization technique. A modified offset-current linearization scheme is proposed to avoid such discontinuity. Behavioral simulations for a 10 GHz ΔΣ-FDC based PLL show that the proposed scheme reduces the spurious tones level more than 10 dB, achieving a worst-case in-band spur level below −54 dBc and σ_{JT} below 80 fs.

ACKNOWLEDGEMENTS

This chapter, in part, is currently being prepared for submission for publication of the material. A. I. Eissa, I. Galton. The dissertation author is the primary investigator and author of this material. Professor Ian Galton supervised the research which forms the basis for this material.

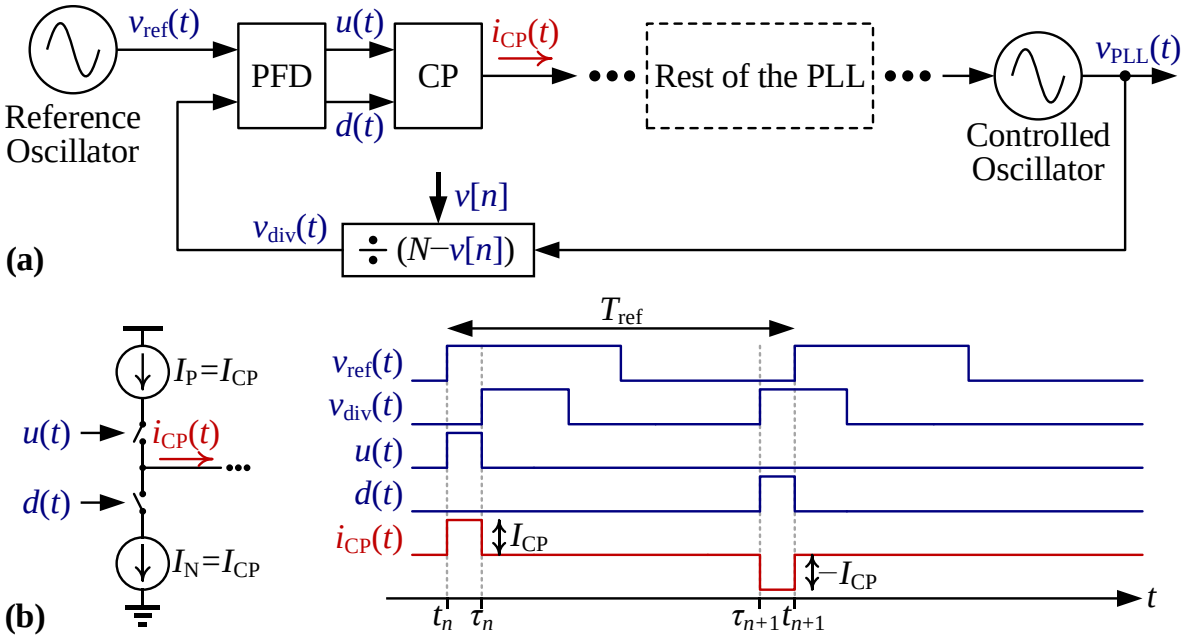


Figure 3.1. (a) Top-level block diagram of a generic fractional-*N* PLL, and (b) conventional PFD and CP operation (ideally, $I_N = I_P = I_{CP}$).

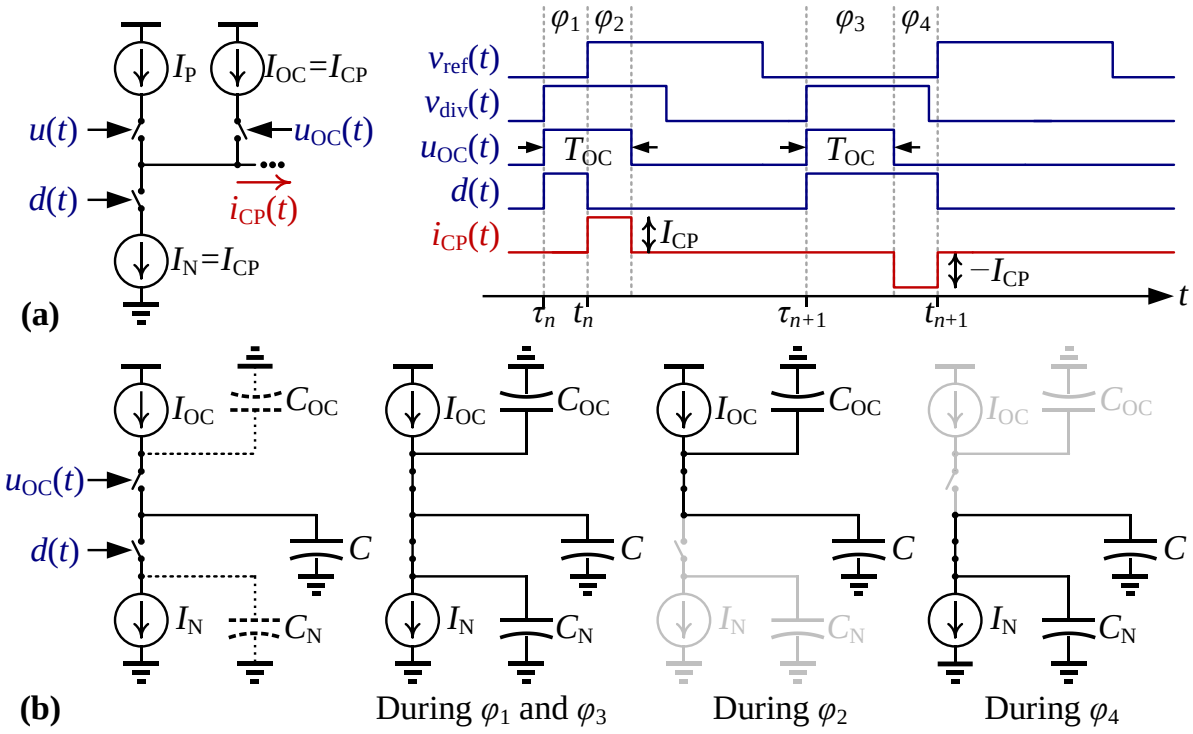


Figure 3.2. (a) Conventional offset-current CP linearization technique (ideally, $I_{OC} = I_N = I_{CP}$), and (b) CP equivalent circuits during the different operation phases $\varphi_{1\text{-}4}$.

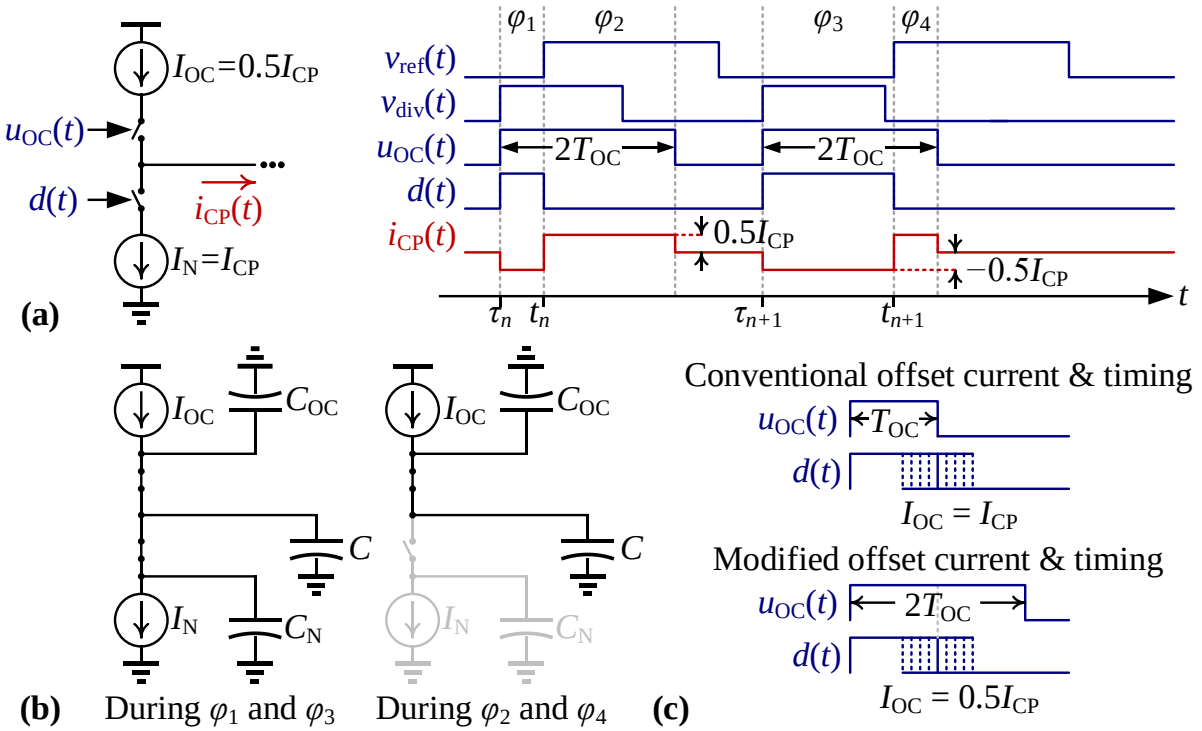


Figure 3.3. (a) Proposed offset-current linearization scheme (ideally, $I_N = I_{CP} = 2I_{OC}$), (b) CP equivalent circuits during $\varphi_{1\text{-}4}$, and (c) conventional vs proposed schemes summary.

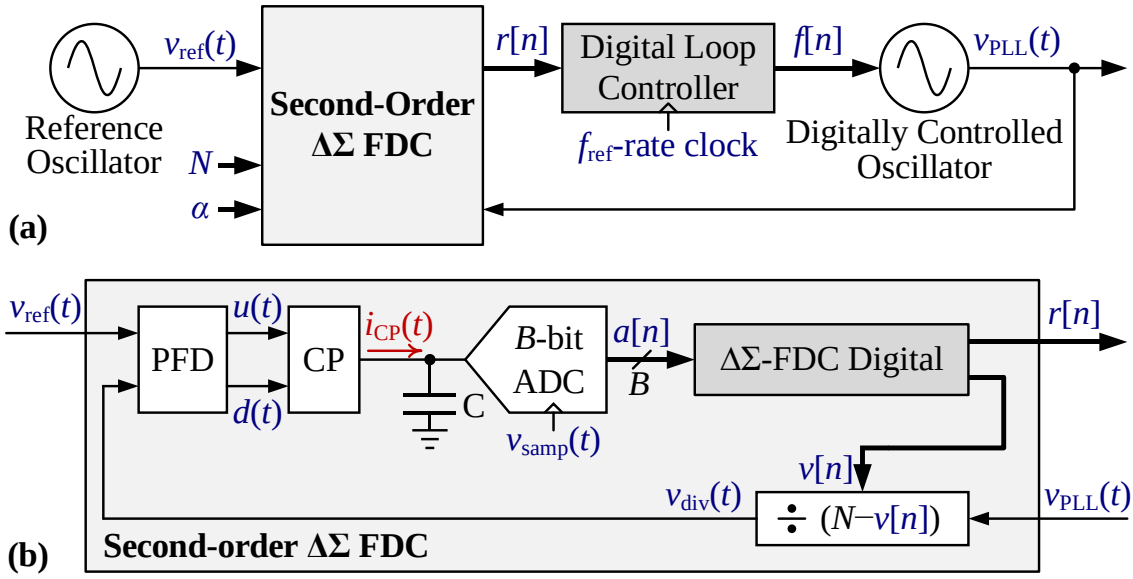
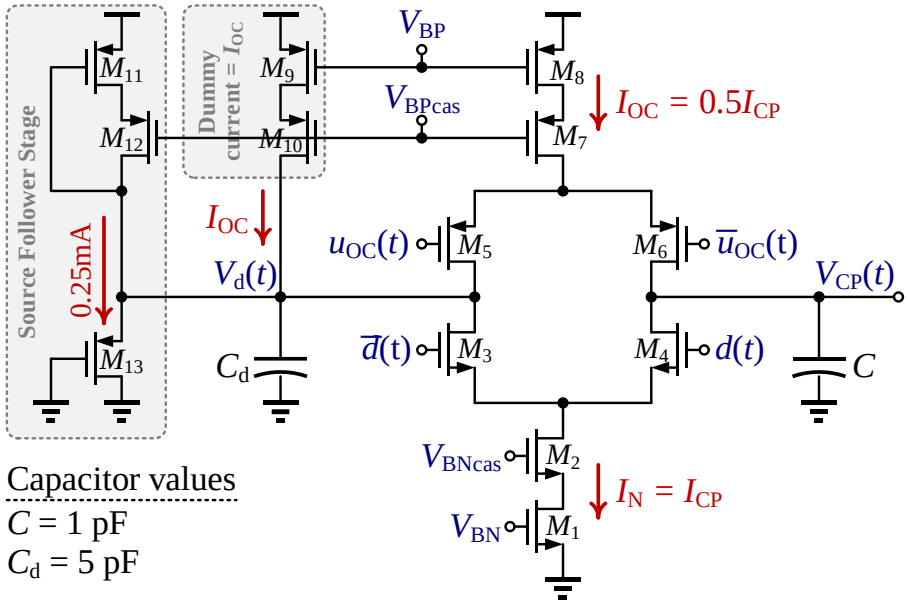


Figure 3.4. (a) A top-level block diagram for the PLL architecture, and (b) simplified block diagram of the ΔΣ-FDC.

	W [µm]	L [nm]
M_1	182.4	280
M_2	19.2	36
$M_{3,4}$	6	20
$M_{5,6}$	12	20
$M_{7,10}$	28.8	36
$M_{8,9}$	320	368
$M_{11,13}$	120	368
M_{12}	14.4	36



Current values
I_{OC} = 0.5 mA
I_{CP} = 1.0 mA

Capacitor values
C = 1 pF
C_d = 5 pF

Figure 3.5. CP circuit implementation and device sizing.

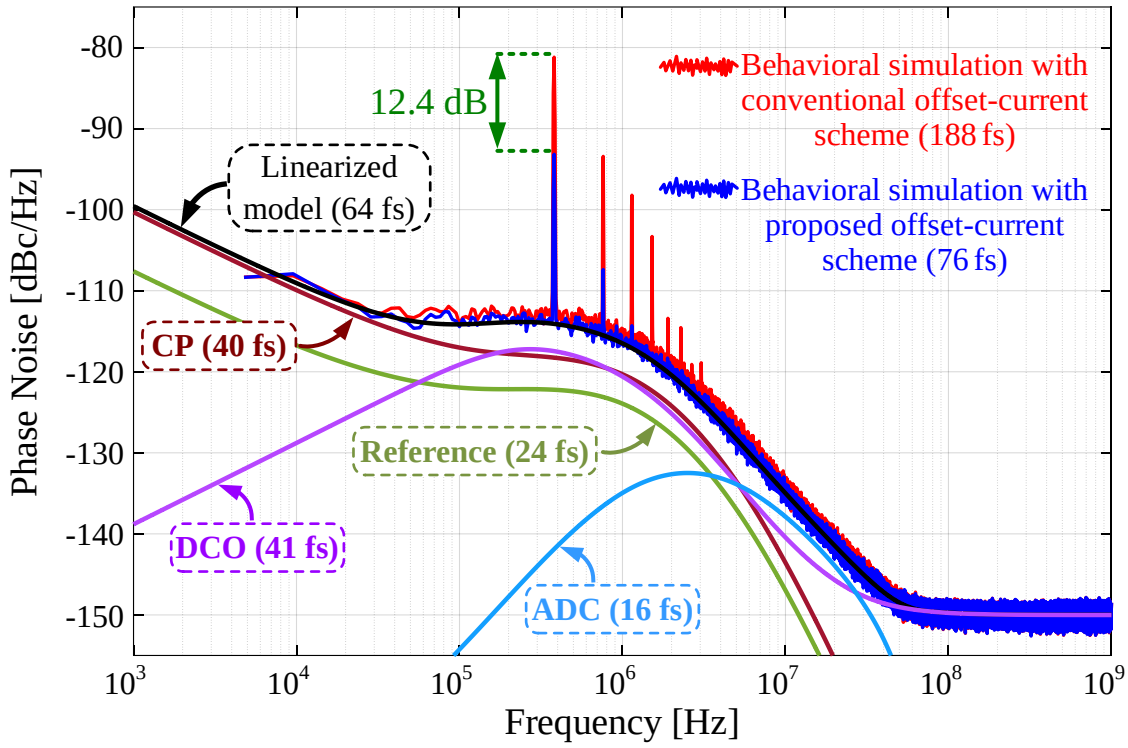


Figure 3.6. PLL's phase noise PSDs with and without the proposed offset current CP linearization schemes, along with individual phase noise contributions estimated from the PLL's linearized model.

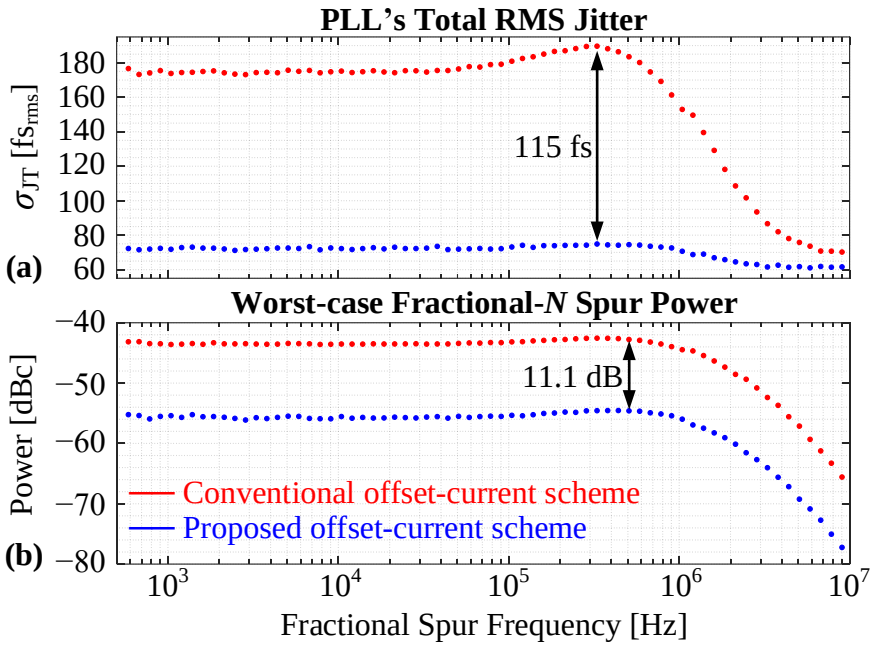


Figure 3.7. (a) Total integrated jitter, σ_{JT}, and (b) power level of the largest measured fractional spur with and without the proposed offset current scheme for fractional frequencies between 500 Hz and 10 MHz.

TABLES

Table 3.1. PLL design parameters used for the behavioral simulations

Design Parameters		Value
Reference Oscillator	Frequency, $f_{crystal}$	153.6 MHz
	Phase noise[1]	−154 and −160 dBc/Hz
Charge Pump	I_{CP} and I_{OC}	1 mA and 0.5 mA
	Bias and buffer current	1.5 mA
	Nominal capacitance, C	1 pF
	Noise[2]	−146 and −156 dBV/Hz
	Supply voltage	0.8 V
ADC	Number of bits	7
	Full-scale voltage	0.4 V
DCO	DCO gain[3]	150 kHz
	Phase noise[4]	−122, −122, and −150 dBc/Hz
PLL Settings	Integer multiplier, N	65
	Fractional multiplier, α	0.002477194
	Output frequency, f_{PLL}	9.98 GHz
	Loop bandwidth	1.71 MHz

[1] $1/f$ and white phase noise components at 10 kHz offset.
[2] $1/f$ and white discrete-time PSD of $e_{CP}[n]$ at 10 kHz offset, where $e_{CP}[n]$ is the noise voltage introduced by the CP across the capacitor C, over the nth reference period.
[3] The DCO gain is defined as the amount by which the DCO frequency changes when its control word changes by unity.
[4] $1/f^3$, $1/f^2$ and white phase noise components at 1 MHz offset.

REFERENCES

1. W. Wu, "Low-Jitter Frequency Generation Techniques for 5G Communication: A tutorial," in *IEEE Solid-State Circuits Magazine*, vol. 13, no. 4, pp. 44-63, Fall 2021.

2. M. Oveisi and P. Heydari, "A Study of BER and EVM Degradation in Digital Modulation Schemes Due to PLL Jitter and Communication-Link Noise," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 8, pp. 3402-3415, Aug. 2022.

3. S. Levantino, "Recent Advances in High-Performance Frequency Synthesizer Design," *2022 IEEE Custom Integrated Circuits Conference (CICC)*, Newport Beach, CA, USA, 2022, pp. 1-7.

4. M. Shinagawa, Y. Akazawa and T. Wakimoto, "Jitter analysis of high-speed sampling systems," in *IEEE Journal of Solid-State Circuits*, vol. 25, no. 1, pp. 220-224, Feb. 1990.

5. B. Razavi, "Jitter-Power Trade-Offs in PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 4, pp. 1381-1387, April 2021.

6. B. Razavi, "Lower Bounds on Power Consumption of Clock Generators for ADCs," *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020, pp. 1-5.

7. I. Galton and C. Weltin-Wu, "Understanding Phase Error and Jitter: Definitions, Implications, Simulations, and Measurement," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 1-19, Jan. 2019.

8. Behzad Razavi, "Delta-Sigma Fractional-*N* Phase-Locked Loops," in *Phase-Locking in High-Performance Systems: From Devices to Architectures*, IEEE, 2003.

9. Bram De Muer and M. S. J. Steyaert, "On the Analysis of ΔΣ Fractional-*N* Frequency Synthesizers for High-Spectral Purity," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 11, pp. 784-793, Nov. 2003.

10. A. Swaminathan, A. Panigada, E. Masry and I. Galton, "A Digital Requantizer With Shaped Requantization Noise That Remains Well Behaved After Nonlinear Distortion," in *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5382-5394, Nov. 2007.

11. E. Familier and I. Galton, "A Fundamental Limitation of DC-Free Quantization Noise With Respect To Nonlinearity-Induced Spurious Tones," in *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 4172-4180, Aug.15, 2013.

12. M. P. Kennedy, V. Mazzaro and D. Mai, "Nonlinearity-Induced Spurs in Fractional-*N* Frequency Synthesizers," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 2617-2622, June 2022.

13. B. Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*. Cambridge: Cambridge University Press, 2020.

14. B. Razavi, "An Alternative Analysis of Noise Folding in Fractional-*N* Synthesizers," *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018.

15. E. Temporiti, G. Albasini, I. Bietti, R. Castello, and M. Colombo, "A 700 kHz bandwidth ΣΔ fractional synthesizer with spurs compensation and linearization techniques for WCDMA applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1446–1454, Sep. 2004.

16. S. E. Meninger and M. H. Perrott, "A 1-MHZ bandwidth 3.6-GHz 0.18-/spl mu/m CMOS fractional-*N* synthesizer utilizing a hybrid PFD/DAC structure for reduced broadband phase noise," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 966-980, April 2006.

17. K. J. Wang, A. Swaminathan and I. Galton, "Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4 GHz Fractional-N PLL," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2787-2797, Dec. 2008.

18. V. S. Sadeghi, H. Miar Naimi and M. P. Kennedy, "The Role of Charge Pump Mismatch in the Generation of Integer Boundary Spurs in Fractional-N Frequency Synthesizers: Why Worse Can Be Better," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 12, pp. 862-866, Dec. 2013.

19. C. -F. Liang and P. -Y. Wang, "10.8 A wideband fractional-*N* ring PLL using a near-ground pre-distorted switched-capacitor loop filter," *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, San Francisco, CA, USA, 2015.

20. M. P. Kennedy *et al*., "16.9 4.48GHz 0.18μm SiGe BiCMOS Exact-Frequency Fractional-*N* Frequency Synthesizer with Spurious-Tone Suppression Yielding a -80dBc In-Band Fractional Spur," *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, CA, USA, 2019.

21. J. Jiang, T. Yan, D. Zhou, A. I. Karsilayan and J. Silva-Martinez, "A 2.3-3.9 GHz Fractional-*N* Frequency Synthesizer with Charge Pump and TDC Calibration for Reduced Reference and Fractional Spurs," *2021 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, Atlanta, GA, USA, 2021.

22. S. Kalia *et al*., "A Sub-100 Fs RMSjitter 20 GHz Fractional-*N* Analog PLL With a BAW Resonator Based On-Chip 2.5 GHz Reference," in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 5, pp. 1372-1384, May 2022.

23. C. Venerus and I. Galton, "Delta-Sigma FDC Based Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 5, pp. 1274-1285, May 2013.

24. Z. Xu, M. Miyahara and A. Matsuzawa, "Picosecond Resolution Time-to-Digital Converter Using G_m-C Integrator and SAR-ADC," in *IEEE Transactions on Nuclear Science*, vol. 61, no. 2, pp. 852-859, April 2014.

25. C. Venerus and I. Galton, "A TDC-Free Mostly-Digital FDC-PLL Frequency Synthesizer With a 2.8-3.5 GHz DCO," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 2, pp. 450-463, Feb. 2015.

26. C. Venerus and I. Galton, "Quantization Noise Cancellation for FDC-Based Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 12, pp. 1119-1123, Dec. 2015.

27. Z. Xu, M. Miyahara, K. Okada and A. Matsuzawa, "A 3.6 GHz Low-Noise Fractional-*N* Digital PLL Using SAR-ADC-Based TDC," in *IEEE Journal of Solid-State Circuits*, vol. 51, no. 10, pp. 2345-2356, Oct. 2016.

28. L. Wu, T. Burger, P. Schönle and Q. Huang, "A Power-Efficient Fractional-N DPLL With Phase Error Quantized in Fully Differential-Voltage Domain," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 4, pp. 1254-1264, April 2021.

29. Y. Wang *et al*., "An 8-14GHz 180fs-rms DTC-Less Fractional ADPLL with ADC-Based Direct Phase Digitization in 40nm CMOS," *2024 IEEE Custom Integrated Circuits Conference (CICC)*, Denver, CO, USA, 2024.

30. C. -W. Yao *et al*., "A 14-nm 0.14-psrms Fractional-N Digital PLL With a 0.2-ps Resolution ADC-Assisted Coarse/Fine-Conversion Chopping TDC and TDC Nonlinearity Calibration," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 12, pp. 3446-3457, Dec. 2017.

31. M. Talegaonkar *et al*., "A 5GHz Digital Fractional- N PLL Using a 1-bit Delta–Sigma Frequency-to-Digital Converter in 65 nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 9, pp. 2306-2320, Sept. 2017.

32. E. Alvarez-Fontecilla, A. I. Eissa, E. Helal, C. Weltin-Wu and I. Galton, "Delta-Sigma FDC Enhancements for FDC-Based Digital Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 965-974, March 2021.

33. E. Helal, E. Alvarez-Fontecilla, A. I. Eissa and I. Galton, "A Time Amplifier Assisted Frequency-to-Digital Converter Based Digital Fractional-N PLL," in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 9, pp. 2711-2723, Sept. 2021.

34. E. Helal, A. I. Eissa and I. Galton, "DTC Linearization via Mismatch-Noise Cancellation for Digital Fractional-N PLLs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 12, pp. 4993-5006, Dec. 2022.

35. B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York: McGraw Hill Education, 2017.

36. A. I. Eissa, E. Alvarez-Fontecilla, C. Weltin-Wu and I. Galton, "A Duty-Cycle-Error-Immune Reference Frequency Doubling Technique for Fractional-*N* Digital PLLs," under review in *IEEE Transactions on Circuits and Systems I: Regular Papers*.

CHAPTER 4

A 75 fs 9–11 GHz ΔΣ-FDC PLL IC: SYSTEM ARCHITECTURE REVIEW

In this chapter, the system architecture details of a Delta-Sigma Frequency-to-Digital (ΔΣ-FDC) Converter based phase-locked loop IC are presented. The presented PLL incorporates the techniques presented in chapters two and three of this dissertation, and targets a total RMS jitter of 75 fs with an FoM of –249 dB. This target places the PLL's performance ahead of state-of-the-art digital PLLs and in-line with state-of-the-art analog PLLs. Table 4.1 summarizes the PLL's target specifications, and the list below summarizes the PLL's key innovations relative to prior art:

1) Calibration-free reference frequency doubling
2) Charge pump static-linearity enhancement
3) Relaxed FDC feedback timing
4) Analog-to-digital converter span reduction
5) Modified multi-modulus divider control scheme.

I. FDC-PLL ARCHITECTURE OVERVIEW

The proposed ΔΣ-FDC PLL architecture is shown in Fig. 4.1. It consists of four main components: a reference frequency doubler (RFD), a second-order ΔΣ-FDC, a digital loop controller (DLC), and a digitally-controlled oscillator (DCO).

A crystal oscillator (XO) is used to generate the periodic waveform, $v_{\mathrm{ref}}(t)$, with nominal frequency, f_{ref}. A reference frequency doubler (RFD) is used to generate the waveform, $v_{\mathrm{RFD}}(t)$, with nominal frequency, $f_{\mathrm{RFD}} = 2f_{\mathrm{ref}}$. The PLL generates a periodic output signal, $v_{\mathrm{PLL}}(t)$,

with a nominal frequency f_{PLL}. The ΔΣ-FDC output sequence, $r[n]$, is a measure of the PLL's frequency-error over the nth RFD period plus other noise terms incurred during the frequency-to-digital conversion process. The sequence $r[n]$ is then accumulated to obtain a phase-error estimate as in conventional ΔΣ PLLs. The estimated phase-error passes through a loop filter (LF) within the DLC, and the LF output, $d[n]$, is used to control the DCO's frequency.

Prior to locking, the PLL's output frequency can be expressed as $(N + \alpha')f_{RFD}$ where α' is, in general, not equal to the parameter α input to the ΔΣ-FDC in Fig. 4.1. The ΔΣ-FDC output, $r[n]$, will change in a direction that forces α' to converge to α. After lock is acquired, the PLL's average output frequency, f_{PLL}, is equal to $(N + \alpha)f_{RFD}$ and its instantaneous output frequency, in Hz, is:

$$f_{PLL}(t) = f_c + K_{DCO} d[n-1] + \psi_{DCO}(t), \tag{72}$$

where $\psi_{DCO}(t)$ is its instantaneous frequency error in Hz, and K_{DCO} is the DCO's gain[6]. The PLL's instantaneous output phase, in cycles, relative to an initial time, t_0, is the integral of (72) from time t_0 to time t:

$$p_{PLL}(t) = (t - t_0)\, f_{PLL} + \theta_{PLL}(t), \tag{73}$$

where

$$\theta_{PLL}(t) = \int_{t_0}^{t} \psi_{PLL}(u)\, du \tag{74}$$

is the PLL's instantaneous phase noise in cycles and $\psi_{PLL}(t)$ is its instantaneous frequency error in Hz.

[6] The DCO gain is defined as the amount by which the DCO frequency changes when its control word, $d[n]$, changes by unity.

As in conventional fractional-N PLLs, the ΔΣ-FDC's output control the PLL's output frequency such that $\psi_{PLL}(t)$ has zero mean and the power spectral density (PSD) of $\theta_{PLL}(t)$ is within acceptable limits for the desired application.

Fig. 4.2(a) shows the top-level block-diagram of the proposed ΔΣ-FDC. It consists of a phase-detector (PD), a charge-pump (CP), an active integrator, a successive approximation register (SAR) analog-to-digital converter (ADC), a multi-modulus divider (MMD) and an FDC digital block. The FDC digital block, shown in Fig. 4.2(b), consists of two discrete-time transfer functions, $R(z)$ and $F(z)$, a coarse digital re-quantizer (Q_C) implemented as a second-order error-feedback delta-sigma modulator (ΔΣM2), a coarse quantization-noise cancellation (QNC) path (adding $c[n]$ to $b[n]$), and a background FDC gain calibration loop. Cancelling the coarse quantization-error after the ADC minimizes the coarse quantization-error contribution to the ADC's span. The FDC gain calibration loop corrects for gain errors in the ΔΣ-FDC forward path and has the form of a standard signed-LMS loop that uses the sign of the quantization-error samples, $e_{qc}[n]$, as a reference signal, and $b[n]$ plus $c[n]$ (the QNC operation output) as the error signal.

The ΔΣ-FDC has four inputs: 1) the periodic waveform $v_{RFD}(t)$ with nominal frequency f_{RFD}, 2) the periodic waveform $v_{PLL}(t)$ with nominal frequency f_{PLL}, 3) an integer-valued modulus, N, and 4) a fractional-valued modulus, α, such that $|\alpha| < 1/2$. The ΔΣ-FDC's output, $r[n]$, is a measure of the PLL's frequency-error over the nth RFD period. Specifically, and assuming all noise sources considered in the system are zero-mean, the mean of $r[n]$ converges to $N + \alpha - (f_{PLL}/f_{RFD})$ after the PLL locks.

The rising edges of $v_{RFD}(t)$ are modulated, in time, by the duty-cycle error of $v_{ref}(t)$. This modulation can be modeled as an alternating error sequence added to the values of the times of the $v_{RFD}(t)$ waveform positive-going zero-crossings had the duty-cycle of $v_{ref}(t)$ been 50%. The transfer functions $R(z)$ and $F(z)$ are designed such that the ΔΣ-FDC preserves its desired input-output transfer function and second-order high-pass ADC quantization error shaping, while perfectly canceling the error in the rising edge times of $v_{RFD}(t)$ at the PD's output. This is achieved by placing a notch at $f = f_{ref}$ ($z = -1$) in the transfer function from the duty-cycle error sequence to the PD output. In the presented design, $R(z)$ and $F(z)$ are given by:

$$R(z) = \frac{1}{\left(1+z^{-1}\right)^2}, \quad \text{and} \quad F(z) = z^{-1}\left(2 - z^{-2}\right). \tag{75}$$

The MMD generates three signals, $v_{div}(t)$, $v_{div\_ext}(t)$, and $v_{conv\text{-}mmd}(t)$. Denoting the time of the nth positive-going zero-crossing of the divider's output by τ_n, the MMD generates the signal $v_{div}(t)$ such that $\tau_n - \tau_{n-1} = (N - v[n-1])T_{PLL}$. The rising edges of $v_{div}(t)$ are compared, within the PD, to those of the RFD output, generating the signal $d(t)$. The other PD output signal, $u(t)$, is a pulse with constant width, T_{OC}, with its rising edge aligned with the rising edge of $v_{div}(t)$. When the PLL is locked, the RFD rising edges lag the MMD rising edges. The signals $v_{div\_ext}(t)$ goes high as $u(t)$ goes low and stays high for a programmable number of PLL cycles. It is used as a ready signal to trigger the digital clock. The $v_{conv\text{-}mmd}(t)$ rising and falling edges mark the start and stop times of the ADC conversion interval, respectively, and are a programmable number of PLL periods referenced to the falling edge of $u(t)$.

The CP uses a current-steering topology with the UP (PMOS) and DN (NMOS) currents steered to the CP output whenever $u(t)$ and $d(t)$ are high, respectively. Otherwise, they are

steered to a dummy branch. The active integrator accumulates the CP dumped charge and generates two output signals, $V_a(t)$ and V_R. Ideally, V_R is equal to a reference voltage level, V_{ref}, and $V_a(t)$ is equal to V_R plus the accumulated CP charge.

When $v_{\text{conv-mmd}}(t)$ is low, the ADC is in the sampling mode with its input terminals connected to the active integrator outputs. The ADC's output is sampled by the FDC digital block and gets normalized by the gain calibration loop coefficient (such that the loop gain is unity). The coarse quantization-error sample is added to the normalized output to perform QNC. The output is clipped and then processed by $R(z)$, $F(z)$, and Q_C to generate $r[n]$ and $v[n]$.

II. CHIP OVERVIEW

Fig. 4.3 shows a top-level block-diagram of the PLL. The chip has five supply domains: 1) vdd_ref_0p8v for the XO and RFD, 2) vdd_fdc_0p8v for all analog blocks within the FDC, 3) vdd_dig_0p8v for all PNR digital blocks, 4) vdd_dco_0p8v for all DCO core blocks and 0.8V buffers, and 5) vdd_drv_1p5v for the DCO output drivers. The chip has two ground domains, vss_dig for the PNR digital block and vss_ana for everything else. The two grounds are connected at the PCB's ground paddle.

The main chip IOs are the supplies (power as described above, and grounds to the paddle), the XO terminals xtal_d and xtal_g, the reference voltage and current I_{ref} and V_{ref}, the PLL outputs, $v_{\text{PLL+}}(t)$ and $v_{\text{PLL–}}(t)$, and the SPI IOs (not shown).

The clock and reset (CNR) block generates the different clocks needed by the PLL's digital circuitry. Fig. 4.4. shows the CNR output main clock domains. 1) clock.dco which is essentially the same as clk_dig_fast, generated by dividing the DCO's output by 10, and is used

to clock registers in the DCO digital circuitry. 2) clock.fdc_dlc.clk which is an f_{RFD}-rate clock whose rising edges are either the RFD signal rising edges or the vdiv_ext rising edges re-synchronized with clk_dig_fast. It is used to clock the FDC digital block, the DLC, and to strobe data from the DCO's digital block. 3) clock.regs, which is a gated version of $v_{RFD}(t)$ that is used to clock the SPI registers.

The ADC receives a clock signal that marks the start and end of the conversion interval. There are two options for such clock generation: 1) using an inverted version of $v_{RFD}(t)$, and 2) using the $v_{conv\text{-}mmd}(t)$ signal generated by the MMD. For the former option, the conversion period is controlled by the RFD period and its pulse width, T_{DL}. For the latter option the clock signal is generated based on the PLL frequency. For N less than or equal to 64 ($\equiv f_{PLL} \leq 9.9072$ GHz), the vconv_mmd signal goes high after five MMD pre-scaler counts ($20T_{PLL}$) and then goes low after an additional five MMD pre-scaler counts ($20T_{PLL}$). For N greater than 64, the vconv_mmd signal goes high after six MMD pre-scaler counts ($24T_{PLL}$) and then goes low after an additional six MMD pre-scaler counts ($23T_{PLL}$ or $24T_{PLL}$). This guarantees that the time allowed for the active integrator to settle (before the ADC's sampling switch is turned OFF) and for the ADC to complete its asynchronous bit-cycling conversion is at least 2ns across the PLL's output frequency tuning range. Fig. 4.5 shows an example timing diagram for the PLL at f_{DCO} = 9.984 GHz.

The ADC output is resampled by the FDC's digital circuitry at the rising edges of clock.fdc_dlc.clk. The FDC digital outputs the phase error sequence, perr, to the DLC and the new divider modulus values, num_div3_phases and num_div4_phases to the MMD. The DLC processes the phase error sequence through a cascade of a proportional-integral stage and a

lowpass IIR stage, requantizes the filtering process output to the target number of bits, and passes the result, fctrl, to the DCO digital. The DCO digital is clocked at the fast rate, clock.dco, and outputs three sequences that control the frequency of the DCO.

Table 4.2 summarizes the PLL parameters.

III. IC ARCHITECTURE DETAILS: ANALOG BLOCKS

A. *Digitally-Controlled Oscillator*

Fig. 4.6 shows the DCO topology. The DCO core is a conventional NMOS LC-oscillator with a tail tank for common-mode resonance [1]. The DCO frequency is adjusted by digitally controlling two capacitor banks; a bank of frequency-control-elements (FCEs) with a coarse frequency step, referred to as the CFCE bank, and a bank of FCEs with a finer frequency step, referred to as the FFCE bank. The need for having an integer and fractional FCE banks is explained in chapter 1, and the coarse and fine FCEs implementation details can be found in chapter 1, [2] and [3]. The CFCE bank is manually controlled through the chip's SPI interface, whereas the FFCE bank elements are controlled by the DLC's output after being processed by the DCO digital controller module. The codeword for the FFCEs integer bank is an f_{RFD}-rate sequence (clock domain = clock.fdc_dlc.clk), while the codeword for the FFCEs fractional bank is an f_{fast}-rate signal (clock domain = clock.dco). Both code words are resynchronized to the fast clock rate, clock.dco, before modulating the FFCE elements.

The DCO core outputs, v_{DCO+} and v_{DCO-}, drive two ac-coupled pseudo-differential buffers that share the same DCO 0.8V supply. The top buffer's output goes to a divider and then to the PLL's output stage (preliminary choice will be an open-drain stage) that has a

separate 1.5V supply. The bottom pseudo-differential buffer output drives the PLL's MMD and clocking circuitry.

A dco_ena register is instantiated in the SPI. When dco_ena is set to zero, the tail transistor gate is pulled down to ground.

Table 4.3 summarizes the DCO's specifications.

B. *Multi-modulus Divider and Clocking*

Fig. 4.7 shows a top-level functional overview of the MMD and clock generation circuitry. The MMD comprises a 3/4 pre-scaler, a combinational logic block, and an FSM. Three other blocks (divide-by-10, synchronizer & delay-line, and ADC clock generation) are used for clocks generation. The FDC passes the target values for the div3 and div4 phases, and configuration bits are loaded from the SPI. The block outputs four signals; clk_dig_fast, vdiv, vdiv_ext, and vconv_mmd. The clk_dig_fast is the fast clock used within the digital and by the DCO digital interface circuitry. The vdiv signal is used by the PD and CP; the PD compares the rising edge of vdiv to the rising edge of $v_{RFD}(t)$ to obtain the $d(t)$ pulse, and the vdiv pulse itself acts as the $u(t)$ pulse. Both $u(t)$ and $d(t)$ control the CP current steering. If ena_vdiv_ext is high, the vdiv_ext signal is generated. It goes high once vdiv goes low and stays high for a programmable time. Finally, the vconv_mmd signal is generated and used by the ADC to mark the start and end of a given conversion process. Fig. 4.8 shows an example timing diagram.

Fig. 4.9 illustrates the timing constraints on the FDC digital and the details of sampling the div3 and div4 phases by the MMD. The timing diagram assumes a worst-case scenario where the vdiv/$u(t)$ pulse width is set to its maximum value, and the resynchronization with clk_dig_fast (to start the FDC digital processing) happens after a full clk_dig_fast cycle. The

MMD's configuration signal, samp_ctrl_delay, defines when to grab the new div3 and div4 phase count. For N (the integer part of the PLL's frequency control word) less than or equal to 64 ($\equiv f_{PLL} \leq 9.9072$ GHz), the MMD sample strobe goes high after 9 counts to 4 (at $36T_{PLL}$ in Fig. 4.9) and stays high until the next pre-scaler edge. The MMD combinational logic and FSM compare the current count4 to the target div4 minus 1. If the result is zero, the MMD starts the div3 count starting the next pre-scaler edge. Otherwise, the pre-scaler continues to divide by 4 until the comparison is zero. This synchronous comparison and update of the pre-scaler modulus control signal guarantees robust operation across various operating conditions. The combinational logic within the MMD, however, must be able to finish its processing in less than 200ps, which is verified across PVT variations via simulations. For N greater than 64 the MMD sample strobe goes high after 10 counts to 4 (at $40T_{PLL}$ in Fig. 4.9) and the same processing above applies.

For N less than or equal to 64 ($\equiv f_{PLL} \leq 9.9072$ GHz), the vconv_mmd signal goes high after five pre-scaler counts ($20T_{PLL}$) and then goes low after an additional five pre-scaler counts ($20T_{PLL}$). For N greater than 64, the vconv_mmd signal goes high after six pre-scaler counts ($24T_{PLL}$) and then goes low after an additional 6 pre-scaler counts ($23T_{PLL}$ or $24T_{PLL}$). This guarantees that the time allowed for the active integrator to settle (before the ADC's sampling switch is turned OFF) and for the ADC to complete its asynchronous bit-cycling conversion is at least 2ns across the PLL's output frequency tuning range.

Table 4.4 summarizes the MMD's specifications.

C. *Crystal Oscillator and Frequency Doubler*

The FDC-PLL requires an f_{RFD}-rate signal, $v_{RFD}(t)$, as an input to the PD, to clock the SPI, as an optional strobe for clocking the PLL's digital circuitry, and as an optional ADC clock. In this PLL design, f_{RFD} is equal to 153.6 MHz, and is generated by passing a reference signal, $v_{ref}(t)$, through an RFD. The signal $v_{ref}(t)$ can be generated by using a 76.8 MHz XO or by using an external source. Fig. 4.10 shows how these two operation modes are be realized (manual setting by adding/removing the jumpers).

Fig. 4.11 shows the reference oscillator implementation details. The rise_trim and fall_trim buses are used to control the reference waveform duty-cycle. The ena_test signal is set high when we want to pass the signal $v_{ref}(t)$ outside the chip through a test multiplexer (TMUX). Sampled phase noise (PN) simulations show that driving I_2 with xtal_d, rather than xtal_g, results in a better PN performance because of the sharper edges in xtal_d.

Fig. 4.12 shows the RFD implementation details. The rise_trim and fall_trim buses are used to further control the impact of the reference waveform duty-cycle, and the width_trim bus is used to control the duration of the $v_{RFD}(t)$ waveform HIGH period, T_{DL}. The ena_test signal is set high when we want to pass the signal $v_{RFD}(t)$ to the chip's TMUX. The ena_dbl default value is nominally set to 1 and may be set to zero for debugging purposes.
Tables 4.5 and 4.6 summarize the XO's and RFD's specifications, respectively.

D. *Phase Detector*

The PD outputs the signal $d(t)$, which is a pulse with width equal to the different between the times of the RFD and MMD rising edges. When the PLL is locked, the RFD rising edges

lag the MMD rising edges. Fig. 4.14 shows the PD implementation. Table 4.7 summarizes the PFD's specifications.

E. *Charge Pump and Active Integrator*

The CP is based on a current steering core to enable fast current settling. The $u(t)$ and $d(t)$ pulses steer the UP (PMOS) and DN (NMOS) currents, respectively, to the output when high. Fig. 4.15 shows the CP topology and Fig. 4.16 show example time-domain waveforms for the control pulses and CP output current. Nominally, the PMOS current, I_P, is one-half the NMOS current, I_N, and I_N = 1 mA. Table 4.8 summarizes the CP's specifications.

Fig. 4.17 shows the active integrator implementation details. The CP output current is accumulated over the capacitor C_F, and a replica servo loop is used to generate the reference voltage, V_R. A pseudo-differential topology is chosen such that any noise from the supplies, V_{ref} and the servo loop show up as a common-mode noise in $V_a(t)$ and V_R; thereby, cancelled by the differential ADC that follows. Table 4.9 summarizes the active integrator's specifications.

F. *Analog-to-Digital Converter*

Fig. 4.18 shows the ADC top-level architecture. The implementation assumes a differential 7-b asynchronous SAR ADC with top-plate sampling. The ADC samples the active integrator output, $V_a(t)$ against V_R. Once vconv goes high, the sampling switches are OFF, and the conversion starts. The end of conversion is forced by the falling of vconv, or by an internal end-of-conversion flag. Trim bits are available to control the ADC step size, cfixed_trim, and to control the delay through the asynchronous timing loop, asynch_del_trim. After locking, the ADC's input signal range is halved; which implies that b5 = inv(b6). The lock_flag signal is

manually set to high when the PLL is locked. It is passed to the ADC in case we wanted to exploit that in the design. In the ADC's implementation, the capacitance DAC that samples V_R (CDAC-N) does not switch during conversion. This guarantees that common-mode dependent nonlinearity is avoided as the comparator terminals will both converge towards V_R. Table 4.10 summarizes the ADC's specifications.

IV. IC ARCHITECTURE DETAILS: DIGITAL BLOCKS

A. *Overview*

The chip has two main digital parts: 1) the divider's FSM and logic, and 2) the PLL's digital. The divider's FSM and logic are custom designed, whereas the PLL's digital is synthesized. Fig. 4.19 shows the top-level functional block diagram for the PLL's digital that includes the FDC digital, DLC, DCO digital, CNR, SPI, and registers. Fig. 4.20 shows the digital top IOs, grouped by block/functionality, and the respective port clock domain.

B. *Clocking and Reset*

The CNR block top-level block-diagram with its main IOs is shown in Fig. 4.21. The CNR block has three clock inputs, clk_xosc = vrfd, clk_dig_fast, and vdiv_ext. The CNR also has the global reset pin, rstb_pin, that is controlled manually from off-chip as an input, as well as the spi_drvb signal as an SPI-ON indicator signal, and some registers passed from the SPI. The CNR block generates three main clocks; clock.dco is the fast clock rate used to clock the DCO fine FCEs, clock.fdc_dlc.clk used to clock the FDC digital and DLC circuitry, and

clock.regs used to clock the SPI registers. It also generates the reset signals to the four main digital blocks; namely, rstb_fdc, rtsb_dlc, rstb_dco, and rstb_regs.

The figures below illustrate the relevant signals, modes of operation, and configuration bits for the CNR module. The names in blue are descriptive names for different wires (matching the RTL code) and the names in green represent members in hwif_pll_from_regs.cnr (__cnr_regs__out_t) and hwif_pll_to_regs.cnr (__cnr_regs__in_t).

Fig. 4.22 shows the reset signals generation circuitry along with example waveforms. Fig. 4.23 shows the different clock domains generation details. In addition to the three main clocks mentioned in this section's introductory paragraph, there are two more clock signals that are used by the FDC's gain calibration loop; clock.fdc_dlc.sma.clk that is a frequency-divided version of clock.fdc_dlc.clk and clock.fdc_dlc.sma.strobe that is a frequency-divided version of clock.fdc_dlc.sma.clk. These two signals are used to clock the internal registers in the FDC's gain calibration loop and provide a knob to down-sample the loop's activity to save power. Fig. 4.24 shows the waveforms of the different clock domains.

Table 4.11 summarizes the CNR's digital block input, output, and configuration signals.

C. *FDC Digital*

Fig. 4.25 shows the signal-processing details of the FDC's digital block. The FDC's digital samples the ADC output data and performs the required functions to output the phase-error sequence to the DLC, and the count3/4 values to the MMD. Internally, it contains a signed LMS loop whose output coefficient is used to correct for the FDC's forward gain error.

Fig. 4.26 shows the FDC's digital blocks functional implementation details. The figure illustrates the relevant signals, modes of operation, and configuration bits. The names in blue

are descriptive names for different wires (matching the RTL code) and the names in green represent members in hwif_pll_from_regs.fdc (__fdc_regs__out_t) and hwif_pll_to_regs.fdc (__fdc_regs__in_t). The clocks in Fig. 4.26 are referenced to clock.fdc_dlc and rstb is = reset.rstb_fdc.

Conventional divider designs assign the majority count to the smaller pre-scaler count value, i.e., in this case, the majority of the pre-scaler counts would have been counts-to-three. To lower the average operating frequency of the MMD, and hence its power consumption, the pre-scaler count values (calculated in the fbdiv_ctrl block in Fig. 4.26) are computed such that the majority counts are count-to-four. Fig. 4.27 shows a snippet from the System-Verlog (SV) code that shows how the count3/4 values are calculated given a MMD divcode.

Table 4.12 summarizes the FDC's digital block input, output, and configuration signals.

D. *Digital Loop Filter*

The DLC passes the phase-error sequence from the FDC's digital through a cascade of a standard PI-stage and low-pass IIR stage. The loop filter output is digitally re-quantized, and the output is passed to the DCO control logic. The transfer function of the loop filter is given by

$$L(z) = \left(0.125 \times km\right) \times 2^{kp} \left(\frac{2^{-ka}}{1-(1-2^{-ka})z^{-1}} + 2^{-15+ki\_kp} \frac{z^{-1}}{1-z^{-1}} \right) \times \left(\frac{2^{-kr}}{1-(1-2^{-kr})z^{-1}} \right). \quad (76)$$

Table 4.13 summarizes the DLC's digital block input, output, and configuration signals.

E. *DCO Digital*

Fig. 4.28 shows the signal-processing details of the DCO's digital control block. The output of the DLC, $d[n]$, is an f_{RFD}-rate sequence that controls the PLL's DCO frequency. It is split into integer and fractional sequences, $d_I[n]$ and $d_F[n]$, through the integer-boundary avoider (details in [3]). The sequence $d_I[n]$ is converted to its segmented form via the binary-to-segmented encoder and the output, $c_I[n]$, controls the integer FCE bank in Fig. 4.28. The sequence $d_F[n]$ is digitally re-quantized by a second-order delta-sigma modulator and is encoded in a thermometer format using the DEM encoder. The DEM encoder output, $c_F[n]$, controls the fractional FCE bank in Fig. 4.28.

Fig. 4.29 shows the DCO digital blocks functional implementation details. The figure illustrates the relevant signals, modes of operation, and configuration bits. The names in blue are descriptive names for different wires (matching the RTL code) and the names in green represent members in regs_in = hwif_pll_from_regs.dco (__dco_mod_regs__out_t) and regs_out = hwif_pll_to_regs.dco (__dco_mod_regs__in_t). The rstb is = reset.rstb_dco.
Table 4.14 summarizes the DCO's digital block input, output, and configuration signals.

V. BEHAVIORAL SIMULATIONS

Figures 4.30 through 4.40 show behavioral simulation results for the presented PLL architecture. Fig. 4.30 and 4.31 show the clock.fdc_dlc and ADC clocking options. Fig. 4.32 through 4.35 shows the FDC's gain calibration block clock waveforms and convergence results. Fig. 4.36 shows the theoretical vs simulated minimum and maximum divider's modulus values for 10 PLL runs. Fig. 4.37 through 4.40 show the PLL's jitter, gc_coeff, gc_coeff mean error,

ADC span, and DCO fctrl span for 500 PLL runs with random {PLL target frequency, CP gain error, initial XO phase}. The four figures correspond to different digital and ADC clocking signals. In all simulations, the XO's duty-cycle error was set to 8%.

ACKNOWLEDGEMENTS

The author would like to thank Colin W. Wu for setting up the simulation platform and providing the SystemVerilog files used to generate the behavioral simulations.

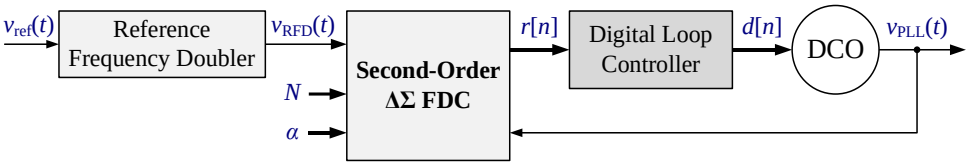


Figure 4.1. Top-level block-diagram of the proposed ΔΣ-FDC PLL architecture.

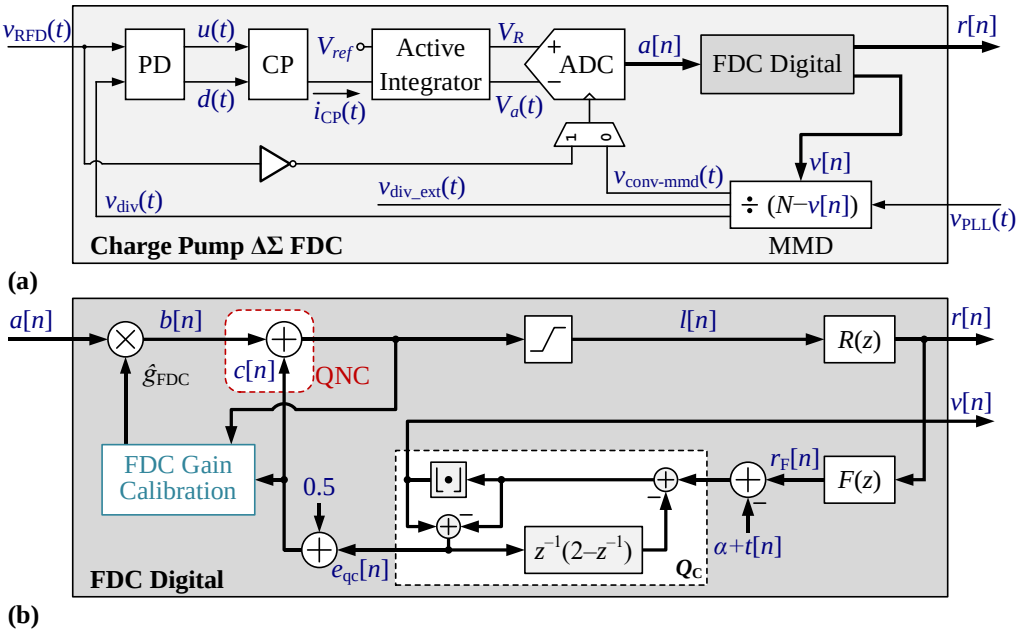


Figure 4.2. (a) Top-level block-diagram of the proposed ΔΣ-FDC, and (b) FDC digital details.

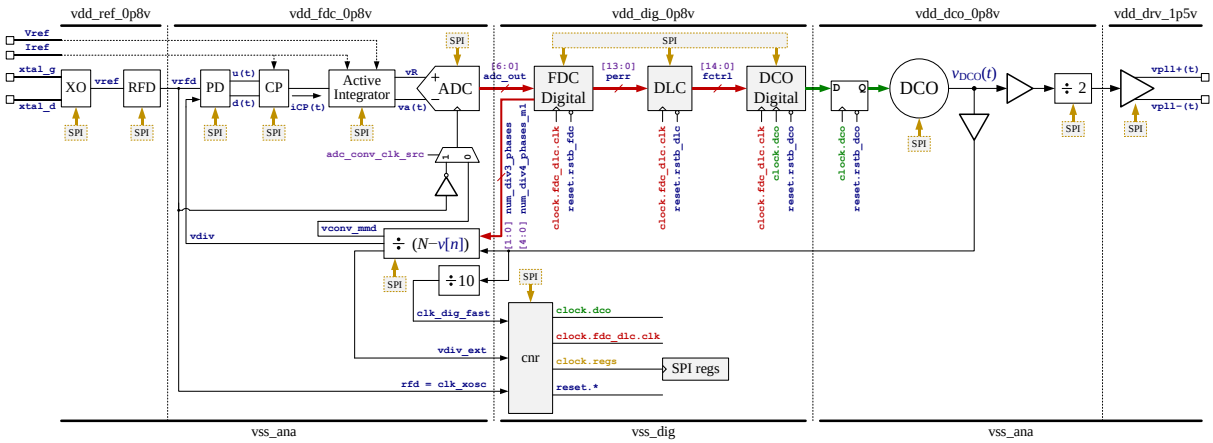


Figure 4.3. PLL's top-level block-diagram.

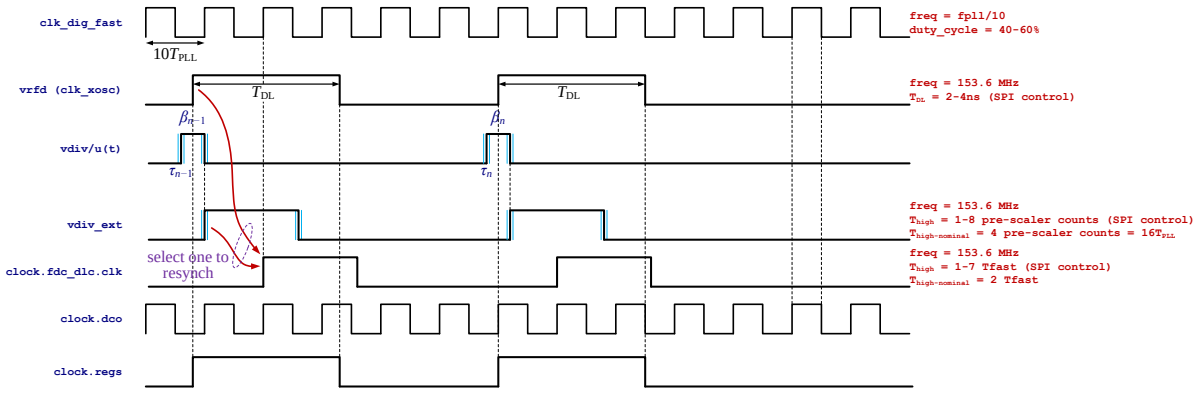


Figure 4.4. PLL's CNR block main clock domains.

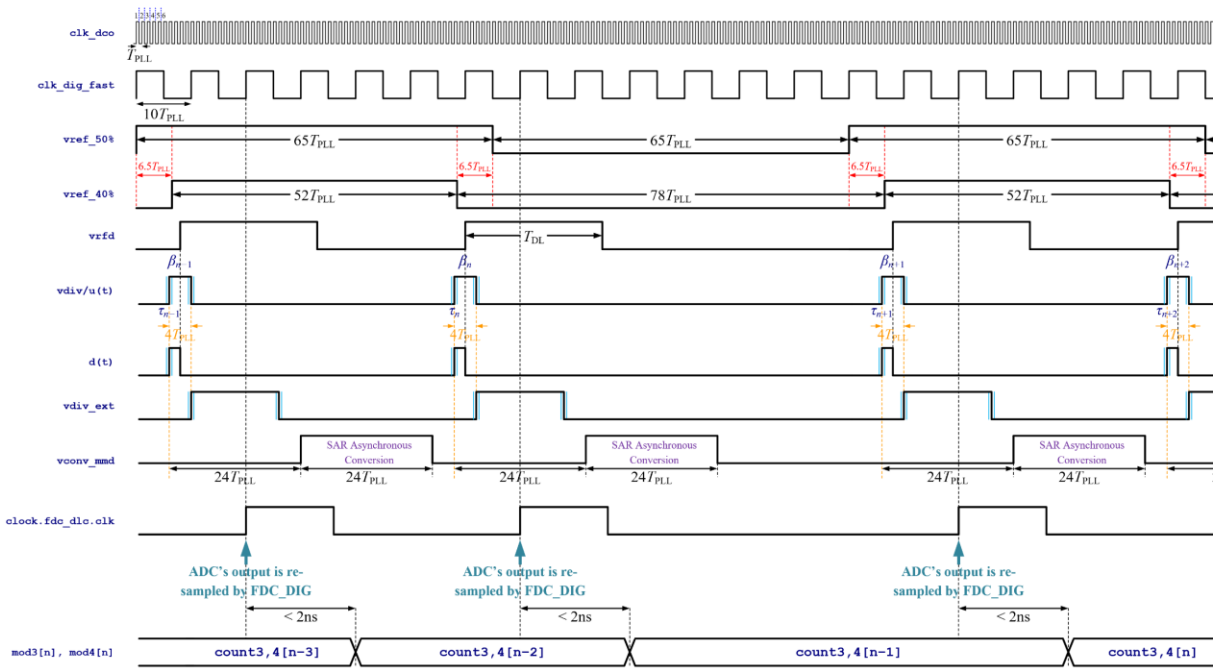


Figure 4.5. Example timing diagram for the PLL at f_{DCO} = 9.984 GHz.

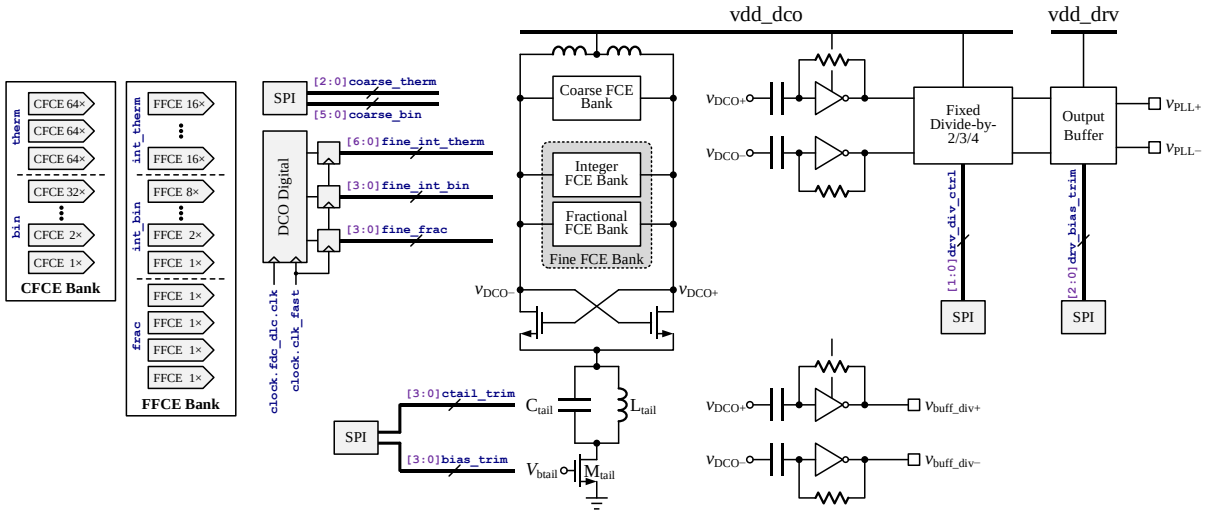


Figure 4.6. DCO topology.

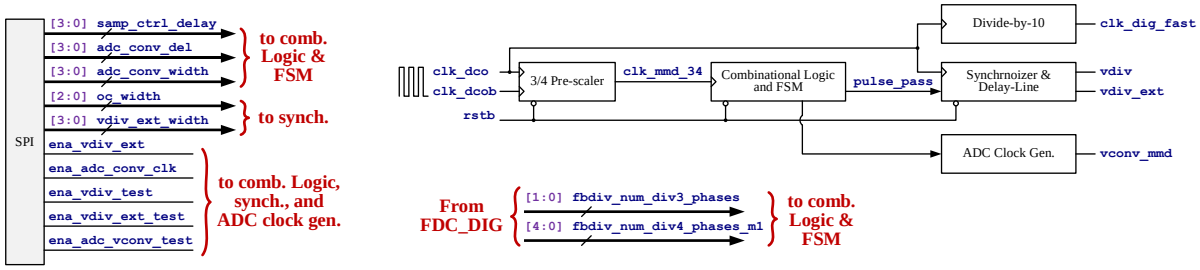


Figure 4.7. Top-level block-diagram for the MMD and clock generation circuitry.

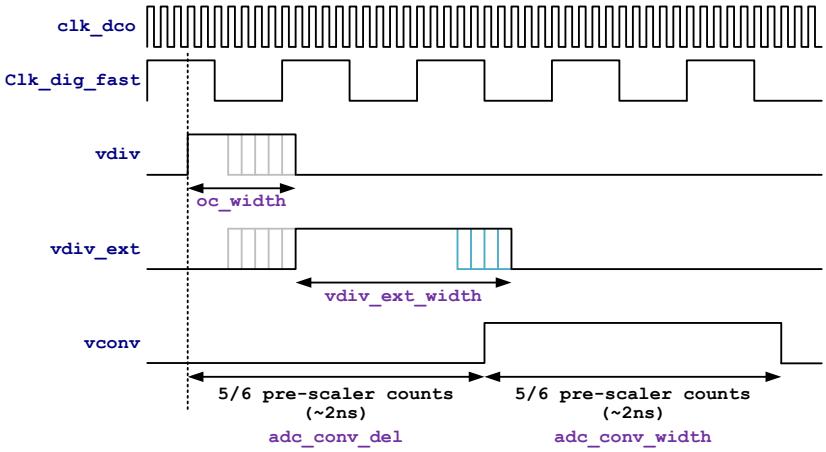


Figure 4.8. Example output waveforms from the MMD and clock generation circuitry.

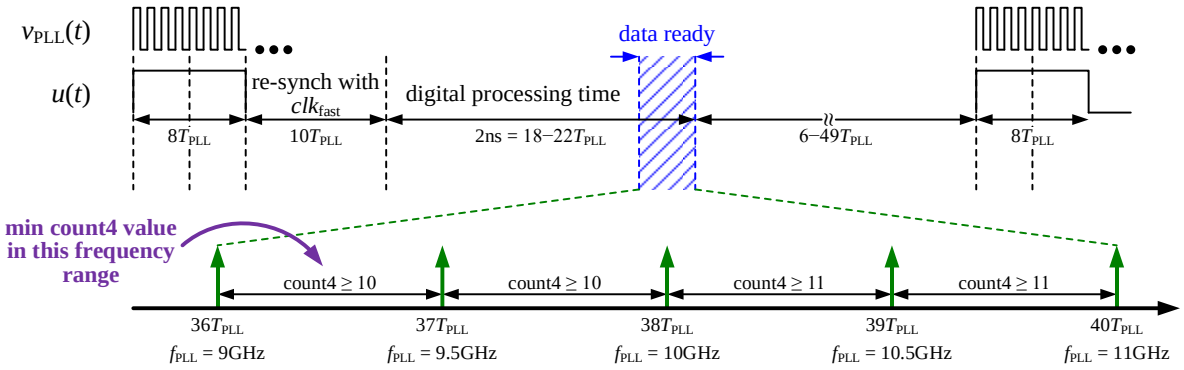


Figure 4.9. Timing diagram illustrating the timing constraints on the FDC digital and MMD data sampling operation.

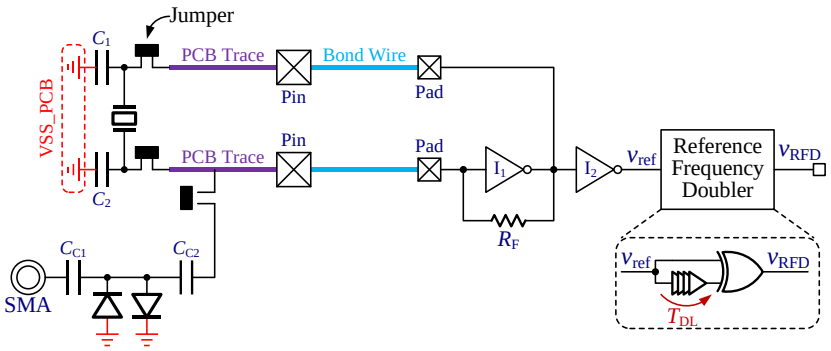


Figure 4.10. Reference signal generation configuration modes.

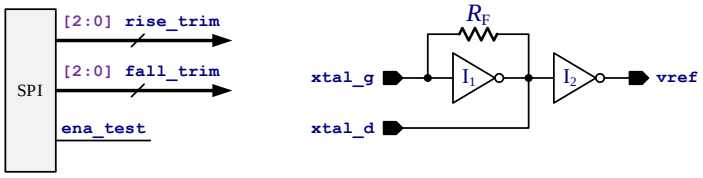


Figure 4.11. Crystal oscillator's Gm stage and reference buffer schematic.

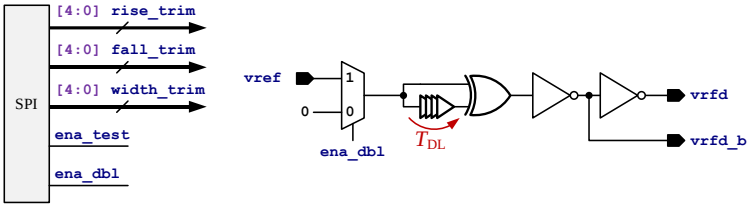


Figure 4.12. RFD top-level diagram.

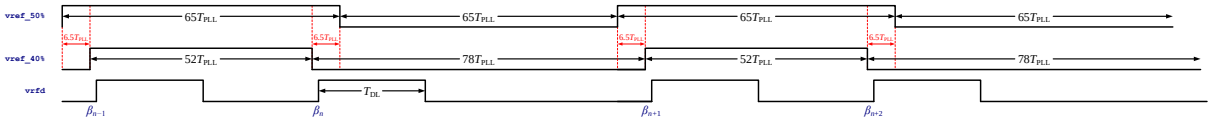


Figure 4.13. Example waveforms for $v_{ref}(t)$ and $v_{RFD}(t)$ with f_{ref} = 76.8MHz and f_{PLL} = 9.984GHz.

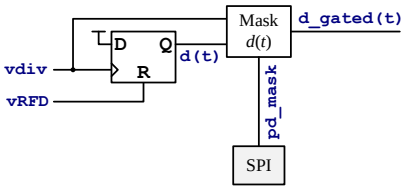


Figure 4.14. PD implementation.

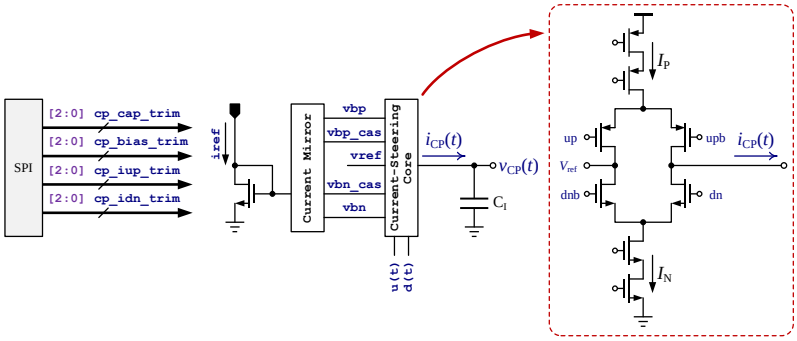


Figure 4.15. CP implementation.

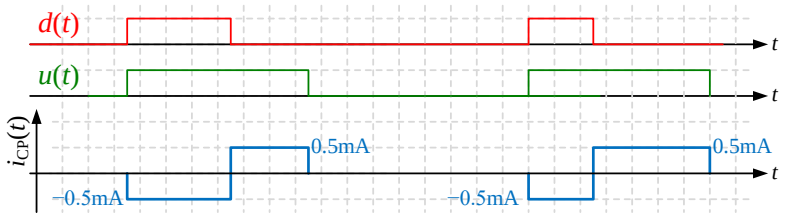


Figure 4.16. CP control pulses and output current example waveforms.

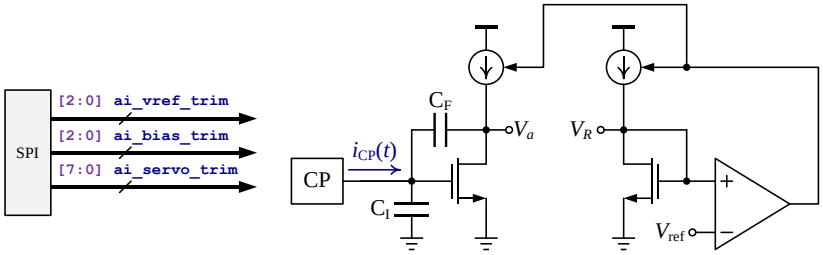


Figure 4.17. Active integrator implementation details.

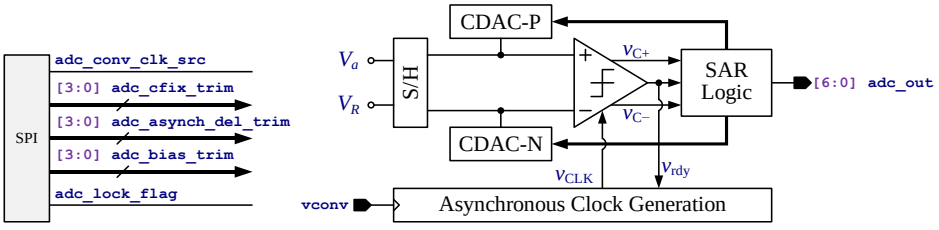


Figure 4.18. 7-bit asynchronous top-sampling SAR ADC top-level architecture details.

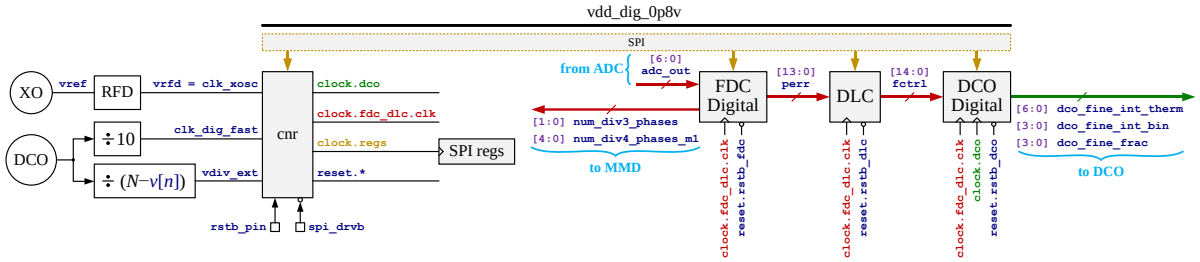


Figure 4.19. PLL's main digital blocks, IOs, and clocking.

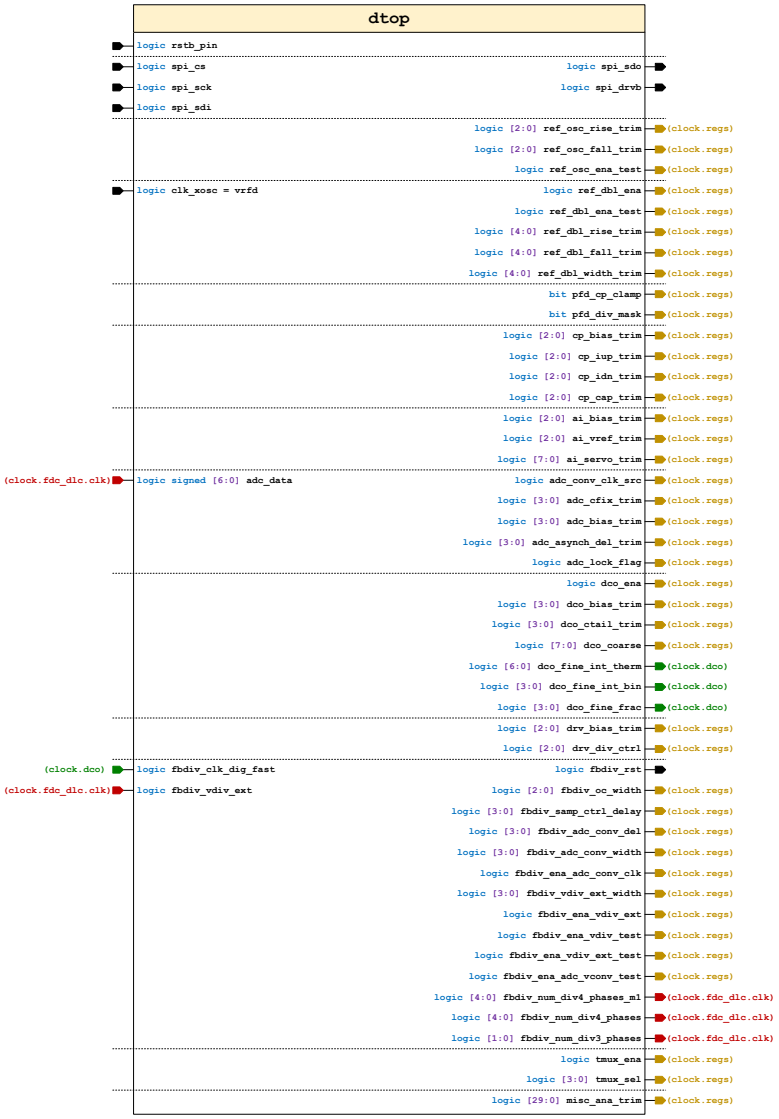


Figure 4.20. the digital top IOs, grouped by block/functionality, and the respective port clock domain.

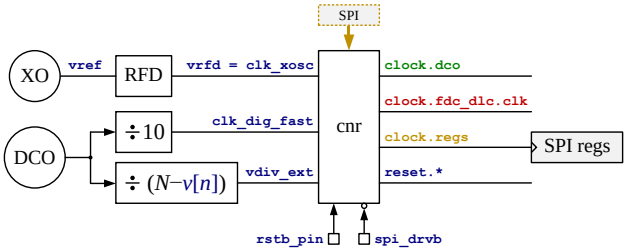


Figure 4.21. CNR top-level block-diagram.

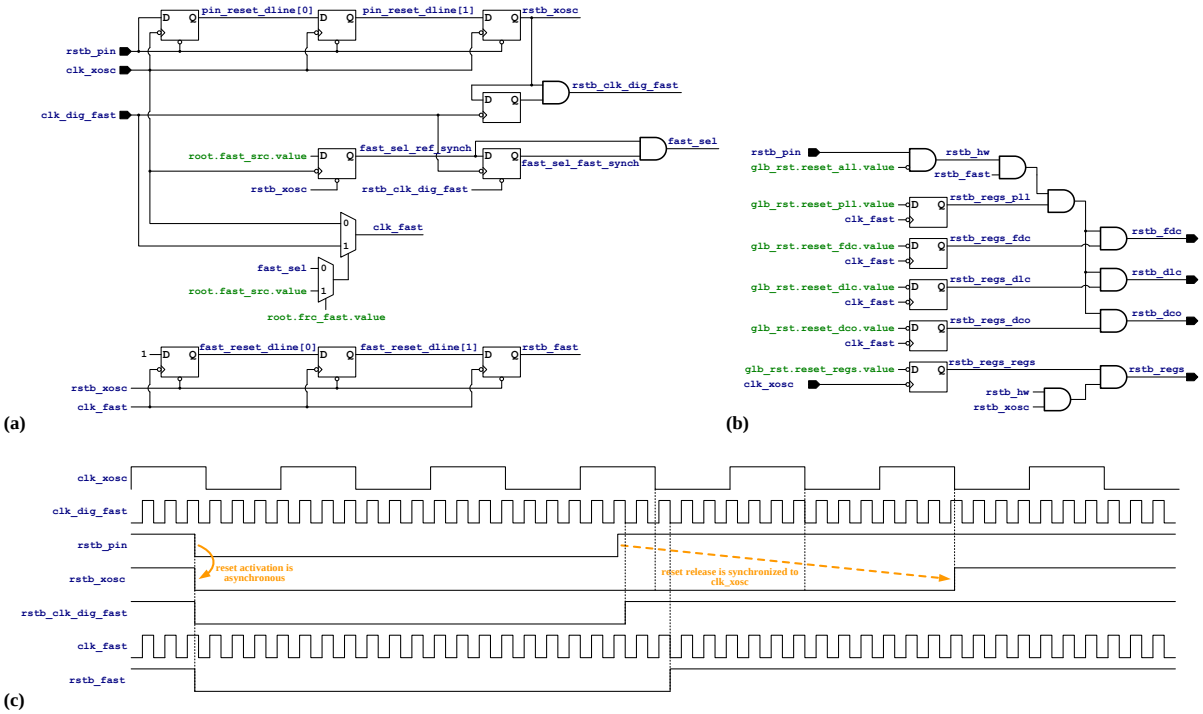


Figure 4.22. (a) Local reset signals generation and fast clock source selection, (b) Reset signals generation for different digital sub-blocks, and (c) Example waveforms showing the asynchronous reset application and its synchronous release.

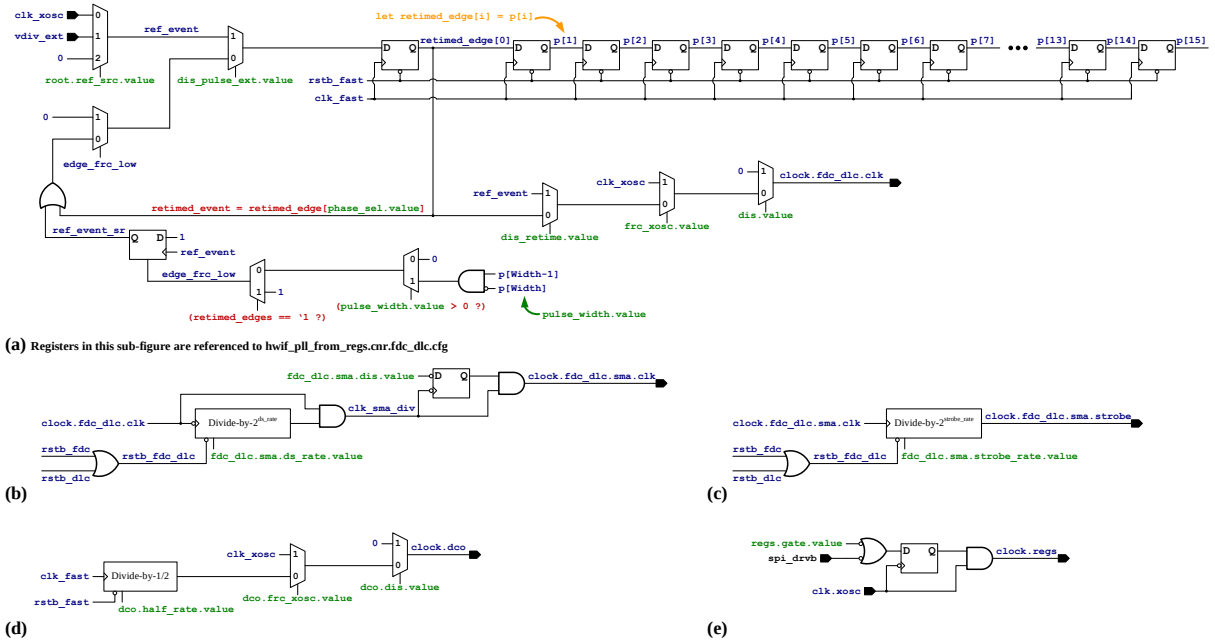


Figure 4.23. Generation details of (a) clock.fdc_dlc, (b) clock.fdc_dlc.sma.clk, (c) clock.fdc_dlc.sma.strobe, (d) clock.dco, and (e) clock.regs.

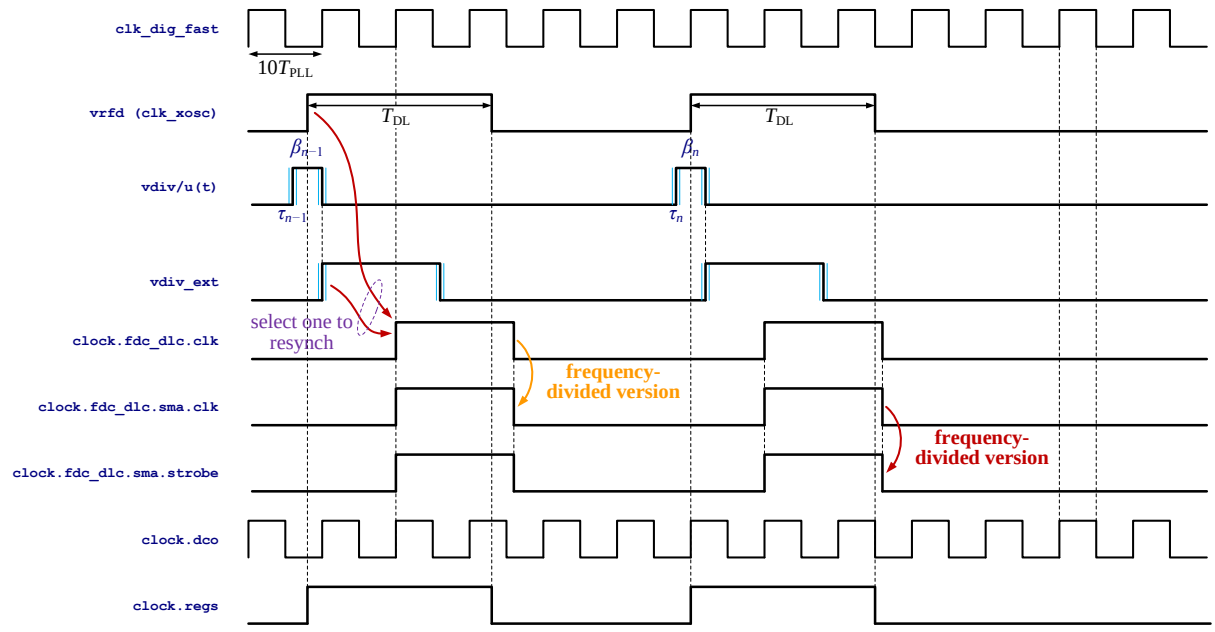


Figure 4.24. Example waveforms for the different generated clocks from the CNR module.

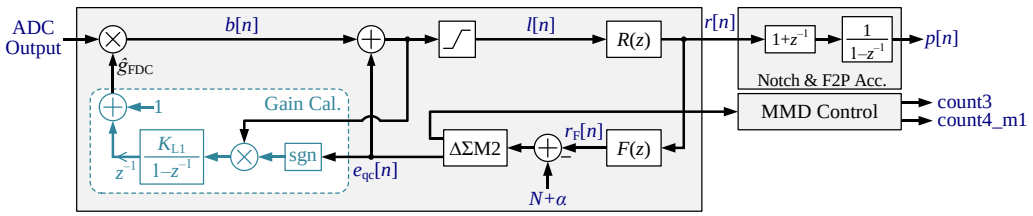


Figure 4.25. FDC signal processing details.

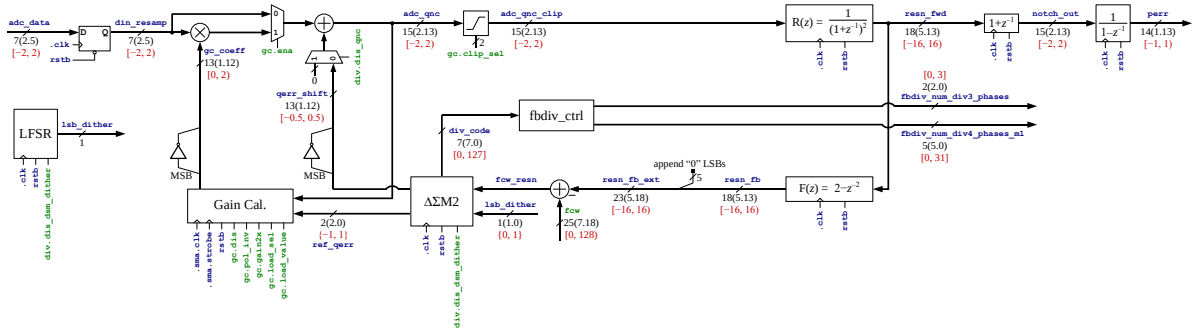


Figure 4.26. FDC digital functional implementation details. N(u,w) indicates a bus-width of N bits interpreted as having u integer bits and w fractional bits.

```
always_comb begin
    mmd_debug_flag = 0;
    if(divcode[1:0] == 0) begin
      num_div3_phases = 0;
      num_div4_phases = divcode[NB-1:2];
      num_div4_phases_m1 = divcode[NB-1:2] - 1;
    end else begin
      num_div3_phases = 4 - divcode[1:0];
      num_div4_phases = divcode[NB-1:2] - num_div3_phases + 1;
      num_div4_phases_m1 = divcode[NB-1:2] - num_div3_phases;
    end
    if(num_div4_phases < MIN_DIV4) begin
      num_div3_phases = 0;
      num_div4_phases = MIN_DIV4;
      num_div4_phases_m1 = MIN_DIV4;
      mmd_debug_flag = 1;
    end
end
```

Figure 4.27. MMD count3 and count4 values calculation. In this design NB = 7 and MIN_DIV4 is 10.

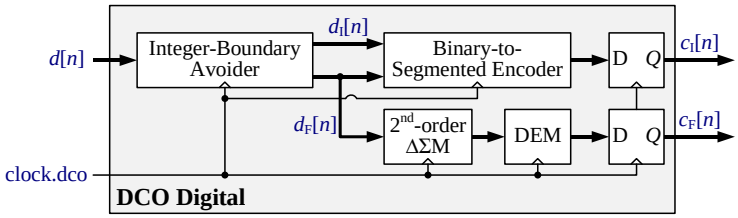


Figure 4.28. DCO digital control signal processing details.

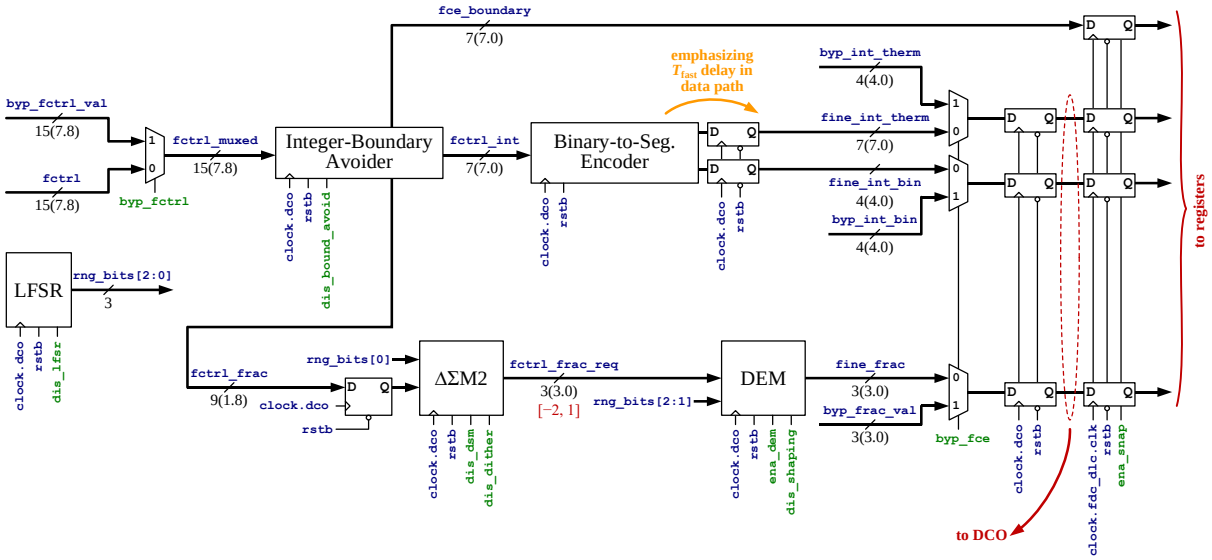
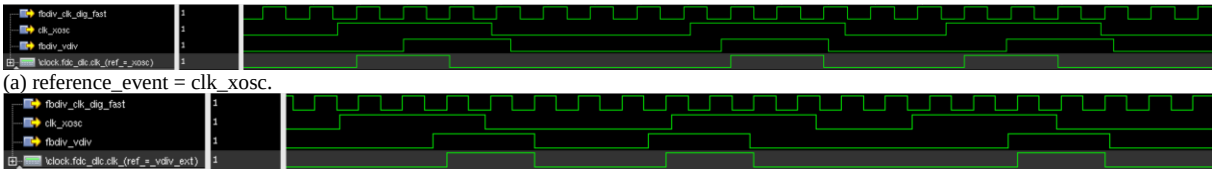


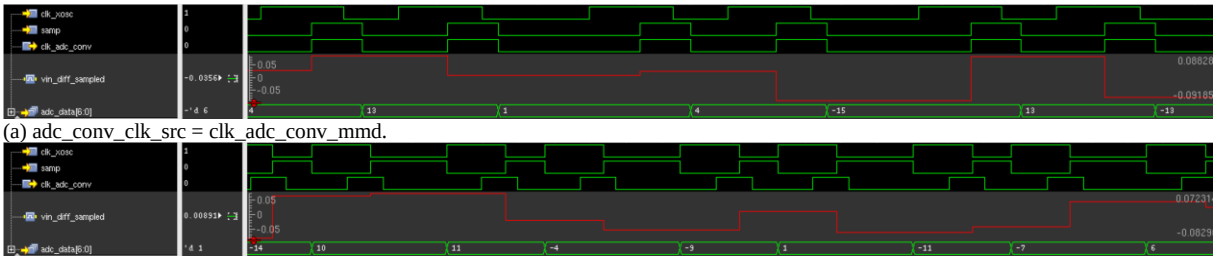
Figure 4.29. DCO digital functional implementation details. $N(u,w)$ indicates a bus-width of N bits interpreted as having u integer bits and w fractional bits.



(a) reference_event = clk_xosc.

(b) reference_event = vdiv_ext.

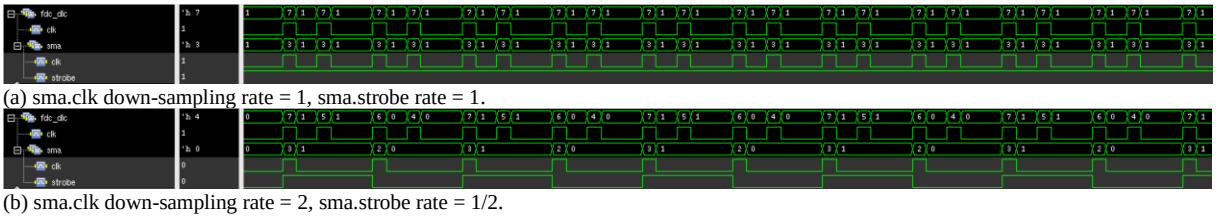
Figure 4.30. Example waveforms showing clock.fdc_dlc generation.



(a) adc_conv_clk_src = clk_adc_conv_mmd.

(b) adc_conv_clk_src = clkb_xosc.

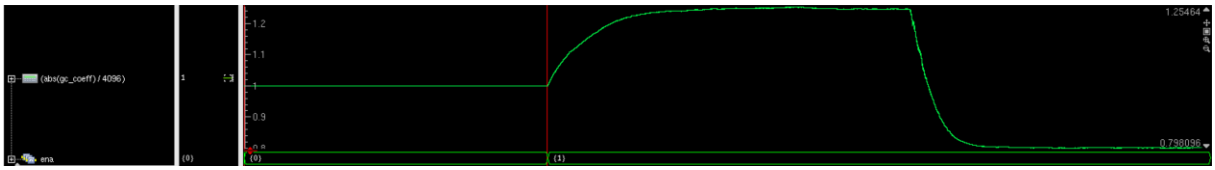
Figure 4.31. Example waveforms showing the two ADC clocking options.



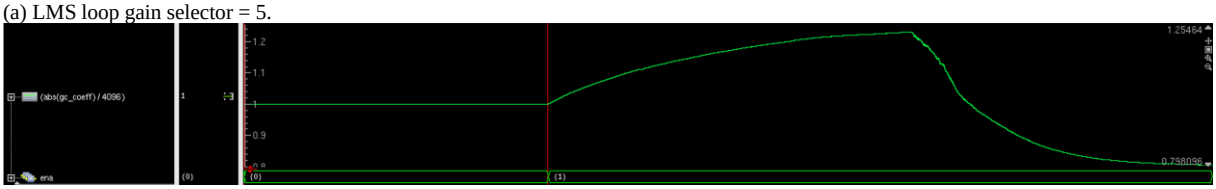
(a) sma.clk down-sampling rate = 1, sma.strobe rate = 1.

(b) sma.clk down-sampling rate = 2, sma.strobe rate = 1/2.

Figure 4.32. Example waveforms showing the FDC gain calibration clocking signals.

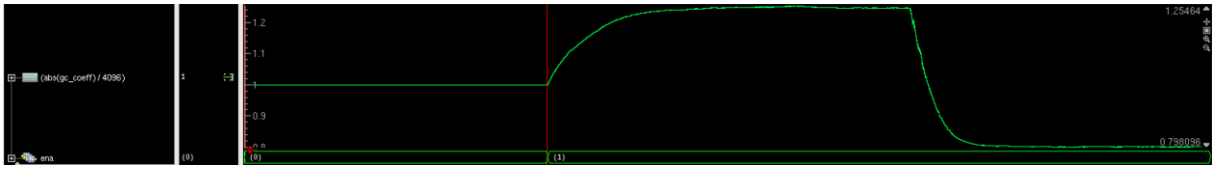


(a) LMS loop gain selector = 5.

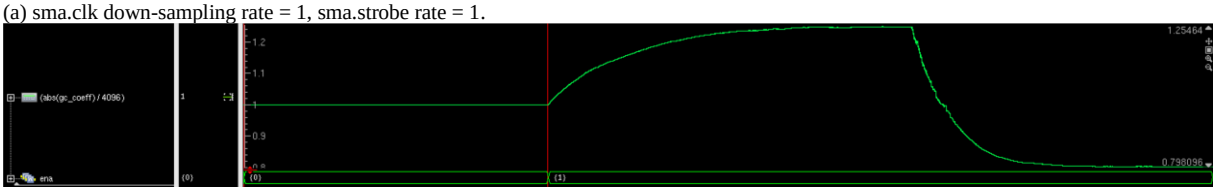


(b) LMS loop gain selector = 4.

Figure 4.33. Example waveforms showing the convergence of the FDC's gain calibration loop coefficient under a step change in the CP current for different LMS loop gains.



(a) sma.clk down-sampling rate = 1, sma.strobe rate = 1.



(b) sma.clk down-sampling rate = 2, sma.strobe rate = 1/2.

Figure 4.34. Example waveforms showing the convergence of the FDC's gain calibration loop coefficient coefficient under a step change in the CP current for different clocking options.

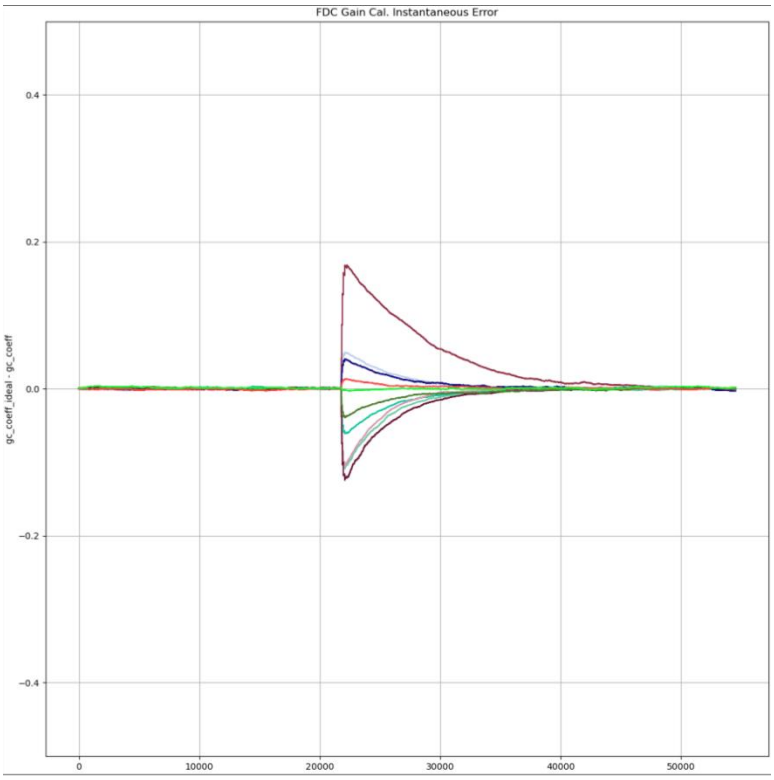


Figure 4.35. Example waveforms showing the FDC's gain calibration loop coefficient error for random CP NMOS current steps (10 PLL runs).

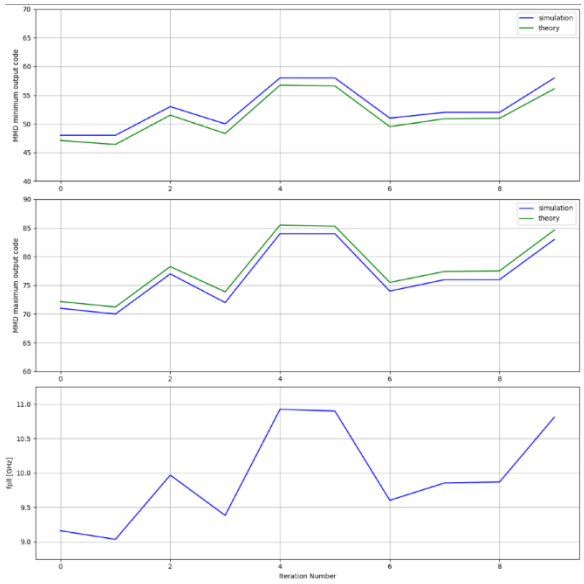


Figure 4.36. Theoretical vs simulated minimum and maximum divider's modulus (10 PLL runs).

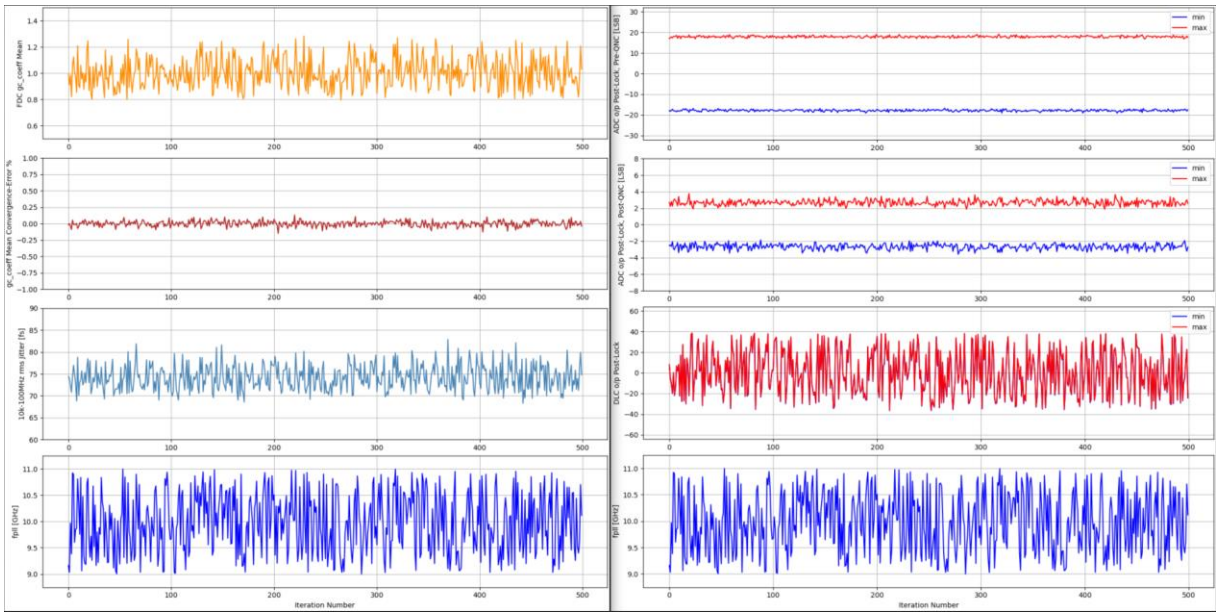


Figure 4.37. Example PLL performance metrics and signals' bounds for reference_event = clk_xosc and adc_conv_clk = vrfd_b.

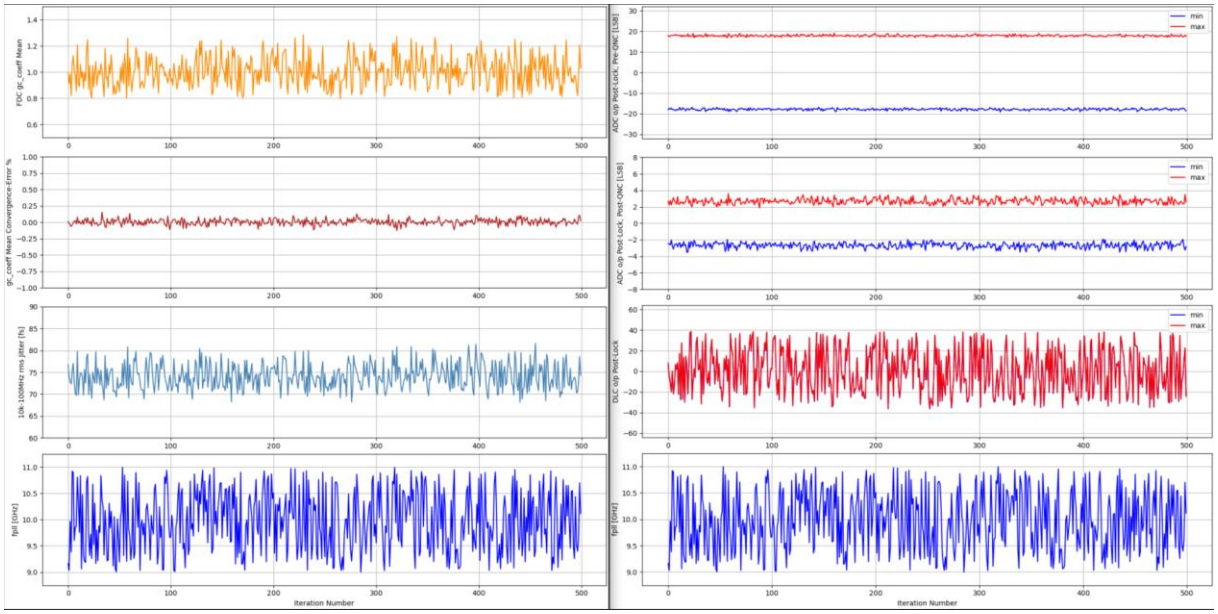


Figure 4.38. Example PLL performance metrics and signals' bounds for reference_event = clk_xosc and adc_conv_clk = adc_conv_clk_mmd.

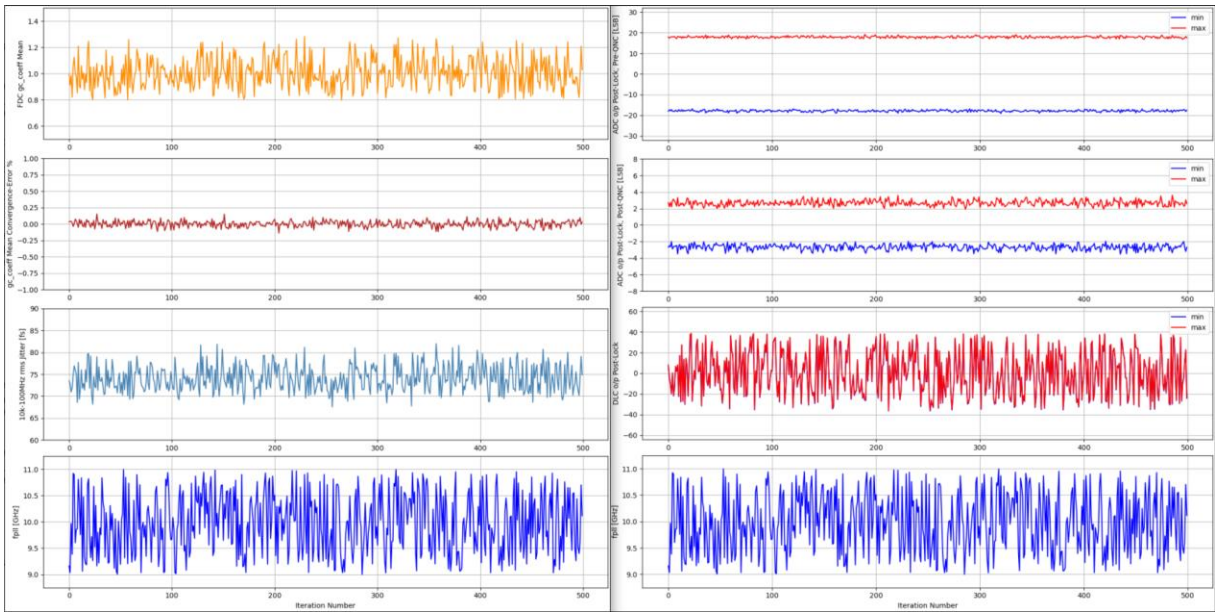


Figure 4.39. Example PLL performance metrics and signals' bounds for reference_event = vdiv_ext and adc_conv_clk = vrfd_b.

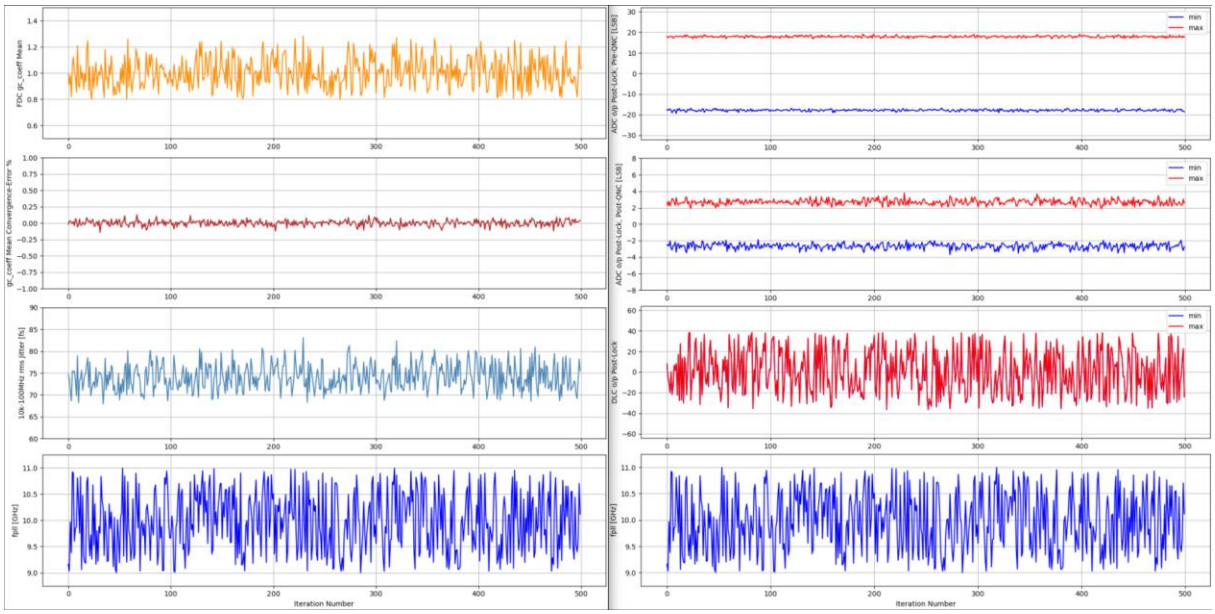


Figure 4.40. Example PLL performance metrics and signals' bounds for reference_event = vdiv_ext and adc_conv_clk = adc_conv_clk_mmd.

TABLES

Table 4.1. PLL target specifications

	This Work
Category	Digital
Architecture	ΔΣ FDC
Technology (nm)	22
Supply (V)	0.8
Area (mm^2)	TBD
f_{ref} (MHz)	76.8
Ref. Freq. Mult.	2
f_{PLL} (GHz)	09.0–11.0
$f_{PLL,report}$ (GHz)	10.00
Total Jitter (fs)	75.00 10k–100M
Fract. Spur (dBc)	−65.0
Ref. Spur (dBc)	−85.0
Power (mW)	18.50
FoM (dB)	−249.8

Table 4.2. Target PLL design parameters and evaluation settings

	Design Parameters	Value	Comments
Reference Source and Frequency Doubler	Crystal frequency	76.8 MHz	
	Reference frequency multiplier	2	
	RFD output white PN	−156 dBc/Hz	Worst-case across PT corners
	RFD output 1/*f*, 10-kHz spot PN	−155 dBc/Hz	Worst-case across PT corners
	Duty-cycle	40–60%	
	Output nominal pulse width	3ns	
	Power consumption	0.5 mW	
DCO	Output frequency range	9–11 GHz	
	Nominal operating frequency	10 GHz	
	Coarse tuning range	2 GHz	
	Coarse frequency step at 10 GHz	15 MHz	Scales with $(f_{DCO}/10G)^3$
	DCO gain, K_{DCO}, at 10 GHz	200 kHz	Scales with $(f_{DCO}/10G)^3$
	$1/f^3$ spot PN at 1 MHz	−125 dBc/Hz	Worst-case across PT corners
	$1/f^2$ spot PN at 1 MHz	−120 dBc/Hz	Worst-case across PT corners
	White phase noise level	−155 dBc/Hz	Worst-case across PT corners
	Power consumption	9.0 mW	
PD	1/*f* spot PN at 10-kHz	−160 dBc/Hz	Worst-case across PT corners Spec for both rising and falling edges
	White phase noise level	−170 dBc/Hz	Worst-case across PT corners Spec for both rising and falling edges
	Power consumption	0.5 mW	
MMD	Modulus range	46–89	
	1/*f* spot PN at 10-kHz	−160 dBc/Hz	Worst-case across PT corners Spec for both rising and falling edges
	White phase noise level	−170 dBc/Hz	Worst-case across PT corners Spec for both rising and falling edges
	Power consumption	1 mW	
ADC	7-bit differential asynchronous SAR ADC		Output interpretation is 2int/5frac
	V_{in+} swing	0–0.8 V	Swing range is 0.2–0.6V after locking
	V_{in-}	0.4 ± 20mV	
	Conversion time	< 2 ns	PLL operation may fail if not met
	Output rate	153.6 MS/s	
	ADC's step-size (≡ 1/ADC's gain)	6.25 mV	
	Gain error	± 2.5%	
	Input-referred offset	± 5 mV	
	Metastability probability of occurrence	< 0.03%	PLL may go out-of-lock if not met
	Sampled noise PSD at ADC's output:		
	1/*f* spot noise at 10-kHz	−135 dBV/Hz	Worst-case across PT corners
	White noise level	−145 dBV/Hz	Worst-case across PT corners
	Linearity performance:		
	CDAC unit cap mismatch	< 2%	
	SFDR	≥ 40 dB	
	Power consumption	1 mW	

	Design Parameters	Value	Comments
CP	Reference current	0.125 mA	
	pMOS nominal current	0.5 mA	
	nMOS nominal current	1.0 mA	
	CP output capacitance, C_P	1.54 pF	
	Average ON time	150ps	
	$e_{CP}[n]$ white floor	−148 dBc/Hz	Sampled noise at CP output
	$e_{CP}[n]$ 1/*f*, 10-kHz spot noise	−138 dBc/Hz	Sampled noise at CP output
	Linearity		Check Active Integrator
	Power consumption	1.25 mW	
Active Integrator	Feedback capacitor	0.5 pF	
	V_{ref}	0.4 V	
	Main OTA:		
	DC gain	TBD	
	GBW	350 MHz	Considering the capacitive FB network
	Servo-loop OTA:		
	DC gain	≥ 40 dB	
	Offset	± 10 mV	
	Output white noise	−150 dBV/Hz	Sampled noise at the output
	Output 1/*f*, 10-kHz spot noise	−135 dBV/Hz	Sampled noise at the output
	Gain error	± 10%	
	Linearity		LUT evaluation for Act. Int. + CP
	Power consumption	1.25 mW	
FDC Digital	FDC GC coefficient		
	Nominal value	1.00	
	Expected range	(0.76, 1.13)	
	Covered range	[0, 2)	
	Timing constraint:		
	From sampling ADC data to div3/4 update	< 2 ns	PLL operation may fail if not met
Loop Dynamics	Loop gain selector, *km*	10	
	Proportional-path gain selector, *kp*	3	
	Integral-path gain selector, *ki_kp*	7	
	Proportional-path IIR pole selector	0	
	IIR pole selector	2	
PLL Settings	Output frequency	9–11 GHz	
	Integer multiplier	58–72	
	Fractional multiplier	–0.5 to 0.5	
	Loop bandwidth	1 MHz	
	Phase margin	75°	
Overall Performance	Total jitter	75 fs	
	Total power consumption	18.5 mW	Assuming 4 mW for digital
	FoM	–249.8 dB	

Table 4.3. DCO specifications summary

Assumptions:

- Target PLL output frequency range is 9–11 GHz.

Supply Network:

- 0.8V supply, vdd_dco_0p8v, for DCO core, digital re-sampling circuitry, and pseudo-differential buffers (3 pins; 6–9 bond-wires).
- 1.5V supply for output drivers (2 pins; 4–6 bond-wires).
- Ground: vss_ana.

Terminals:

Supply	vdd_dco, vdd_drv, vss_ana
Inputs	[6:0]fine_int_therm, [3:0]fine_int_bin, [3:0]fine_frac
Outputs	out_drv and outb_drv (PLL outputs), clk_dco and clkb_dco (to MMD)
Config.	[3:0]dco_bias_trim, [3:0]ctail_trim, [7:0]coarse_bin, [2:0]drv_div_ctrl, [2:0]drv_bias_trim

Loading:

- Output drivers: on-chip routing, pads, package network.
- Bottom inverter: the FDC's MMD and the clock generation circuitry.

Specifications:

- Center frequency: 10 GHz
- Coarse cap. bank:
  - Tuning range = 2 GHz around f_{DCO} = 10 GHz across process and temperature corners
    (! The PLL will not function if the DCO fails to oscillate for at least a narrow range between 9–11 GHz)
  - Coarse frequency step = 15 MHz at f_{DCO} = 10 GHz (scales with $(f_{DCO}/10G)^3$)
- Fine cap. bank:
  - Tuning range = 25.0 MHz at f_{DCO} = 10 GHz (scales with $(f_{DCO}/10G)^3$)
    (! The PLL will not function properly if frequency gaps between coarse steps & fine TR occur)
  - FCE frequency step = 200 kHz at f_{DCO} = 10 GHz (= Δ_{FCE} = K_{DCO}) (scales with $(f_{DCO}/10G)^3$)
- Worst-case phase noise (at the drivers' output) across PLL frequency range, and process and temperature corners
  - Floor = –150 dBc/Hz
  - $1/f^2$ spot PN value at 1 MHz offset = –120 dBc/Hz
  - $1/f^3$ spot PN value at 1 MHz offset = –125 dBc/Hz
- Power consumption budget: 9mW (excluding output path core buffers, divider, and pad drivers)

Table 4.4. MMD specifications summary

Assumptions:

- The DCO started-up successfully and is running in the 9–11 GHz frequency range.

Supply Network:

- 0.8V supply, vdd_fdc_0p8v (5 available pins shared with all other FDC analog components; 10–15 bond-wires).
- vss_ana.

Terminals:

Supply	vdd_div, vss_div
Inputs	clk_dco, clk_dcob, [1:0]fbdiv_num_div3_phases, [4:0]fbdiv_num_div4_phases_m1
Outputs	clock.fast, vdiv, vdiv_ext, and vconv_mmd
Config.	[3:0]samp_ctrl_delay, [3:0]adc_conv_del, [3:0]adc_conv_width, [2:0]oc_width, [1:0]vdiv_ext_width, ena_vdiv_ext, ena_vdiv_test, ena_vdiv_ext_test, ena_adc_vconv_test

Loading:

- ADC sampling switch
- Several logic gates within the ADC's digital and the PLL's digital
- PD
- CP pMOS steering switches

Specifications:

- Modulus range: 46–89 (≡ count4 range: 10–22 and count3 range: 0–3)
- Phase noise (at vdiv rising and falling edges):
  - Floor = –170 dBc/Hz
  - $1/f$ spot PN value at 10-kHz offset = –160 dBc/Hz
- Power consumption budget: 1mW
- Controls ranges and nominal values:
  - oc_width = 3, ∈ [0,8]T_{PLL}
  - vdiv_ext_width = 4 pre-scaler counts, ∈ [0,8] pre-scaler counts
  - samp_ctrl_delay = 8 for N ≤ 64, 9 otherwise
  - adc_conv_del and adc_conv_width = 5 pre-scaler counts for N ≤ 64, 6 otherwise

Table 4.5. Crystal oscillator specifications summary

Assumptions:
- A quart crystal with resonance frequency of 76.8 MHz is available.
- An external signal generator (or off-chip XO) with output frequency of 76.8MHz is available.

Supply Network:
- 0.8V supply, vdd_ref_0p8v (1 available pin shared with RFD; 2–3 bond-wires).
- Ground: vss_ana.

Terminals:
- Supply | vdd_xo and vss_xo
- Inputs | xtal_g, xtal_d
- Outputs | vref
- Config. | [2:0]rise_trim, [2:0]fall_trim, ena_test

Loading:
- The XO output, vref, is loaded by the RFD and the TMUX network.

Specifications:
- Reference frequency = 76.8MHz.
- Output waveform, vref:
  - Duty-cycle range: 40–60% (! The PLL will not function if these limits are violated)
- Phase noise (see RFD spec table).
- Power consumption (see RFD spec table).

Table 4.6. RFD specifications summary

Supply Network:
- 0.8V supply, vdd_ref_0p8v (1 available pin shared with XO; 2–3 bond-wires).
- Ground: vss_ana.

Terminals:
- Supply | vdd_rfd, vss_rfd
- Inputs | vref
- Outputs | vrfd, vrfd_b
- Config. | [4:0]trise_trim, [4:0]tfall_trim, [4:0]width_trim, ena_test

Loading:
- The RFD output, vrfd, is loaded by the PD, the digital clock synchronizer, and the TMUX network.

Specifications:
- Output frequency = 153.6MHz.
- Output waveform, vrfd:
  - Rectangular waveform
  - |tn,rising,ideal – tn,rising,real| < 10% (! The PLL will not function if these limits are violated)
  - Pulse width: 2.0–4.0 ns
- Worst-case phase noise (at vrfd's rising edges) across output frequency range, and process and temperature corners
  - Floor < –156 dBc/Hz
  - 1/*f* spot PN value at 10 kHz offset < –155 dBc/Hz
- Power consumption budget (XO + RFD): 0.5mW (excluding I_1 in Fig. 4.10)

Table 4.7. PD specifications summary

Supply Network:
- 0.8V supply, vdd_fdc_0p8v (5 available pins shared with all other FDC analog components; 10–15 bond-wires).
- vss_ana.

Terminals:
- Supply | vdd_pd, vss_pd
- Inputs | vdiv, vRFD
- Outputs | d_gated
- Config. | pd_mask

Loading:
- CP nMOS steering switches.

Specifications:
- Phase noise (at $d(t)$ rising and falling edges):
  - Floor = –170 dBc/Hz
  - 1/*f* spot PN value at 10 kHz offset = –160 dBc/Hz
- Power consumption budget: 0.5mW

Table 4.8. CP specifications summary

Assumptions:
- A reference current equal to 0.125mA is available.

Supply Network:
- 0.8V supply, vdd_fdc_0p8v (5 available pins shared with all other FDC analog components; 10–15 bond-wires).
- vss_ana.

Terminals:
- Supply | vdd_cp, vss_cp
- Input | $u(t)$ and $d(t)$, V_{ref}
- Output | $i_{CP}(t)$
- Config. | [2:0]cp_cap_trim, [2:0]cp_bias_trim, [2:0]cp_iup_trim, [2:0]cp_idn_trim

Loading:
- CP output capacitance, C_P, and the OTA's equivalent input capacitance.

Specifications:
- pMOS current = 0.5×nMOS current = 1mA
- Total output capacitance = 1.54 pF
- Sampled noise PSD at $v_{CP}[n]$: (transient noise is recommended)
  - Floor = –149 dBc/Hz
  - 1/*f* spot PN value at 10 kHz offset = –138 dBV/Hz
- Power consumption budget: 1.25mW

Table 4.9. Active Integrator specifications summary

Assumptions:
- A reference current equal to 0.125mA is available.
- A reference voltage, V_{ref}, equal to 0.4V is available.
- Current specifications assume a single-pole OTA model.
- OTA's input-referred noise specifications assume CP output resistance of 0.5kΩ

Supply Network:
- 0.8V supply, vdd_fdc_0p8v (5 available pins shared with all other FDC analog components; 10–15 bond-wires).
- vss_ana.

Terminals:
- Supply | vdd_int, vss_int
- Input | $i_{CP}(t)$, V_{ref}
- Output | $v_a(t)$, V_R
- Config. | [2:0]ai_vref_trim, [2:0]ai_int_bias_trim, [7:0]ai_servo_trim

Loading:
- ADC's sampling capacitance (time-varying since connected to the buffer through a sampling switch)
- Feedback network (Feedback capacitor = 0.5pF, CP output capacitor + OTA input capacitor = 1.54pF)

Specifications:
- Gain error[7] < 10%
- Main OTA: GBW ≥ 350 MHz (considering feedback network), input-referred offset[8] < 5mV
- Servo-loop OTA: DC gain > 40dB and offset < ±15mV
- Noise:
  - For the $v_n[n] \times (1 + C_P/C_F)$ term[9]
    - Floor at $v_a[n]$ = –150 dBV/Hz
    - 1/*f* spot value at 10 kHz offset at $v_a[n]$ = –135 dBV/Hz
  - For the $e_{vn,in}[n]$ term[10]
    - OTA's input-referred noise continuous-time PSD floor < –175 dBV/Hz
    - OTA's input-referred noise continuous-time PSD 1/*f* spot value at 10 kHz offset < –150 dBV/Hz
- Linearity[11]
- Power consumption budget: 1 mW

[7]Ideal gain = I_N/C_F. Because of the OTA's finite loop-gain, the actual gain is scaled by $\beta A_0/(1 + \beta A_0)$, where A_0 is the OTA's DC gain and β is the feedback factor.
[8]This can be revisited if the ADC and CP show tolerance to larger offset values.
[9]Recommended testbench = transient noise.
[10]Recommended testbench = ac noise.
[11]The target fractional-*N* spur level is the main metric defining linearity specifications. Linearity should be evaluated by extracting the $d(t)$–$v_a[n]$ characteristics and back-annotating the results in the C code/SV model.

Table 4.10. ADC specifications summary

Assumptions:

- Clock, v_{conv}, is available. It marks the start of the bit-cycling (end-of-sampling) and forces bit-cycling stop.

Supply Network:

- 0.8V supply, vdd_fdc_0p8v (5 available pins shared with all other FDC analog components; 10–15 bond-wires).
- vss_ana.

Terminals:

- Supply: vdd_adc, vss_adc
- Inputs: $v_a(t)$, V_R
- Outputs: [6:0]adc_out (two's complement representation interpreted as 2 integer/5 fractional)
- Config.: [3:0]adc_cfix_trim, [3:0]adc_asynch_del_trim, [3:0]adc_bias_trim, adc_lock_flag

Loading:

- Several gates in the PLL's digital.

Specifications:

- Input signal span:
  - $V_a(t)$ swings from 0 to 0.8V single-ended (after locking, only the 0.2–0.6V range is exercised)
  - V_R = 0.4 ± 20mV (Assumes DC gain > 40dB and offset < ±15mV in the active integrator servo loop OTA)
- ADC step-size (≡ 1/ADC's gain) = 6.25 mV (Example: if V_R = 0.4, and $v_a[n]$ = 0.32, adc_out = 12)
  (! The PLL budgeting allows ±2.5% of gain error variations)
- Sampling rate = 153.6 MHz (Nyquist operation)
- Conversion time ≤ 2 ns (!Data must be ready to be re-sampled by the FDC digital 2ns after the conversion starts. The PLL will not function if this timing constraint is violated)
- CDAC unit capacitance random mismatch < 2%
- Input referred offset < 5mV (not a hard spec, but trades-off with the CP buffer linearity)
- Prob(metastability) < 0.03% (! This is a critical spec as frequent metastable events may drive the PLL out-of-lock)
- Noise (at $v_a[n]$)[12]:
  - Floor = –145 dBV/Hz (SQNR = 43.3 dB, excluding flicker noise))
  - 1/*f* spot value at 10 kHz offset = –135 dBV/Hz
- SFDR > 40 dB (test with full-scale input sinusoid)
- Power consumption budget: 1 mW

Table 4.11. CNR main inputs, outputs, and configuration signals

Summary:

- Inputs: vrfd = clk_xosc, clk_dig_fast, vdiv_ext, rstb_pin, spi_drvb
- Outputs: clock.fdc_dlc.clk, clock.fdc_dlc.sma.clk, clock.fdc_dlc.sma.strobe, clock.dco, clock.regs
  rstb_fdc, rstb_dlc, rstb_dco, rstb_regs
- Config.:

Name	Values	Default	Description
root.fast_src.value	{0,1}	1	Select fast clock source
root.frc_fast.value	{0,1}	0	Forcing clk_fast select
glb_rst.reset_all.value	{0,1}	0	Reset whole chip
glb_rst.reset_pll.value	{0,1}	0	Reset PLL internal states
glb_rst.reset_fdc.value	{0,1}	0	Reset FDC
glb_rst.reset_dlc.value	{0,1}	0	Reset DLC
glb_rst.reset_dco.value	{0,1}	0	Reset DCO
glb_rst.reset_regs.value	{0,1}	0	Reset Registers
fdc_dlc.cfg.root.ref_src.value	{0,1,2}	0	
fdc_dlc.cfg.root.dis_pulse_ext.value	{0,1}	0	Disable pulse extender
fdc_dlc.cfg.phase_sel.value	{0,…,15}	0	Select clk_fast edge to synch with
fdc_dlc.cfg.pulse_width.value	{0,…,7}	2	Select clock pulse width
fdc_dlc.cfg.dis_retime.value	{0,1}	0	Disable retiming and use raw ref_event
fdc_dlc.cfg.frc_xosc.value	{0,1}	0	Force clk_xosc to be used
fdc_dlc.cfg.dis.value	{0,1}	0	Force disable this clock domain
fdc_dlc.sma.dis.value	{0,1}	0	Force disable this clock domain
fdc_dlc.sma.ds_rate.value	{0,…,7}	0	Down-sampling rate selector
fdc_dlc.sma.strobe_rate.value	{0,…,15}	0	Strobing rate selector
dco.half_rate.value	{0,1}	0	Run DCO_dig at half ffast-rate
dco.frc_xosc.value	{0,1}	0	Force clk_xosc to be used
dco.dis.value	{0,1}	0	Force disable this clock domain
regs.gate.value	{0,1}	0	Gate the registers clock

[12]Recommended testbench = transient noise.

Table 4.12. FDC's Digital main inputs, outputs, and configuration signals

Summary:					
▪ Inputs	[6:0]adc_data, rstb used in Fig. 4.26 = reset.rstb_fdc				
▪ Outputs	[13:0]perr, [1:0]fbdiv_num_div3_phases, [4:0]fbdiv_num_div4_phases				
▪ Clocks	The FDC dig uses clock.fdc_dlc.clk, clock.fdc_dlc.sma.clk, and clock.fdc_dlc.sma.strobe				
▪ Timing	The fbdiv_ctrl outputs should be updated at most 2 ns after the ADC data is captured. (! The PLL will not function if this timing constraint is violated)				
	Name	**Values**	**Default**	**Description**	
▪ Config.	gc.dis	{0,1}	0	Enable or disable the update of the GC loop	
	gc.pol_inv	{0,1}	0	Invert the sign of the GC loop input	
	gc.gain2x	{0, …, 7]	5	LMS loop gain coefficient factor selector	
	gc.load_sel	{0,1}	0	Use the user-defined gc_coeff (through the SPI)	
	gc.load_value	N/A	N/A	The user-defined gc_coeff (through the SPI)	
	gc.ena	{0,1}	1	Use the output of the GC loop	
	div.dis_qnc	{0,1}	0	Disable the ΔΣM2 quantization-error QNC	
	gc.clip_sel	{0,1,2,3}	1	Clip the QNC operation output to $\pm 2^{-\text{clip\_sel} + 1}$	
	div.dis_dsm_dither	{0,1}	0	ADD LSB dither to the ΔΣM2 input	

Table 4.13. DLC main inputs, outputs, and configuration signals

Summary:				
▪ Inputs	[13:0]perr, rstb = reset.rstb_dlc			
▪ Outputs	[14:0]fctrl			
▪ Clocks	clock.fdc_dlc.clk			
	Name	**Values**	**Default**	**Description**
▪ Config.	*km*	[0,15]	7	Changes loop filter gain in steps of 0.125
	kp	[0,7]	4	Proportional-path gain
	ka	[0,7]	1	Proportional-path IIR stage pole
	ki_kp	[0,15]	7	Integral-path stage coefficient (norm to *kp*)
	kr	[0,7]	1	IIR stage pole
	byp_perr	{0,1}	0	Use user-defined perr value or use from FDC dig
	intg_path_byp	{0,1}	0	Use user-defined integral path input or use from DLC
	ena_snap	{0,1}	0	Write data to registers

Table 4.14. DCO's Digital main inputs, outputs, and configuration signals

Summary:				
▪ Inputs	[14:0]fctrl, rstb used in Fig. 4.29 = reset.rstb_dco			
▪ Outputs	[6:0]fine_int_therm, [3:0]fine_int_bin, [3:0]fine_frac			
▪ Clocks	clock.dco, clock.fdc_dlc.clk			
	Name	**Values**	**Default**	**Description**
▪ Config.	byp_fctrl	{0,1}	0	Use user-defined fctrl value or use from DLC
	byp_fce	{0,1}	0	Use user-defined fce inputs or use from DCO dig
	ena_snap	{0,1}	0	Write data to registers
	dis_lfsr	{0,1}	0	Enable/disable LFSR
	dis_bound_avoid	{0,1}	0	Enable/disable integer-boundary avoider
	dis_dsm	{0,1}	0	Enable/disable ΔΣM (use uniform quantizer)
	dis_dither	{0,1}	0	Enable/disable the ΔΣM LSB dither
	ena_dem	{0,1}	0	Enable/disable DEM encoder
	dis_shaping	{0,1}	0	Enable/disable using shaped switching-sequences

REFERENCES

1. E. Hegazi, H. Sjoland and A. A. Abidi, "A filtering technique to lower LC oscillator phase noise," in *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 1921-1930, Dec. 2001.

2. C. Venerus and I. Galton, “A TDC-Free Mostly-Digital FDC-PLL Frequency Synthesizer With a 2.8-3.5 GHz DCO,” *IEEE J. Solid-State Circuits*, vol. 50, no. 2, pp. 450–463, Feb. 2015.

3. C. Weltin-Wu, G. Zhao and I. Galton, "A 3.5 GHz Digital Fractional- PLL Frequency Synthesizer Based on Ring Oscillator Frequency-to-Digital Conversion," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 12, pp. 2988-3002, Dec. 2015.