

A Wide Tuning Range All-Digital Phase-Locked Loop with Fine Resolution for Digital Clock Generation in Predictive 7 nm FinFET Technology

Averal Kandala¹ and Micah Roschelle¹

¹Department of Electrical Engineering and Computer Sciences, University of California, Berkeley

Email: averal@eecs.berkeley.edu, micah.roschelle@eecs.berkeley.edu

Abstract — Phase-locked loops (PLLs) are traditionally mixed-signal circuits that play a critical role in clock generation and distribution within SoCs. However, as such systems migrate to more deeply scaled process technologies which are unfriendly towards analog/mixed-signal design, designers have turned to all-digital PLL (ADPLL) designs for their stability over PVT variations, compatibility with automated digital design flows, portability between technologies, and dynamic programmability. In this work, we present an ADPLL design in the ASAP7 7 nm predictive technology suitable for digital clock generation and capable of achieving low jitter or fine output frequency resolution through programming of a modular third-order delta-sigma modulator (DSM) for dithering. Schematic- and system-level test benches verify the operation of the ADPLL over a wide range of operating conditions, with measurements at the nominal supply of 0.7 V showing minimum peak-to-peak jitter and output resolution of 5 ps and 9 MHz, respectively, and a coarse frequency tuning range of 0.499-7.239 GHz, all in a projected die area of 1600 μm^2 with total power consumption of 2.3 mW.

Index Terms – All-Digital Phase-Locked Loop (ADPLL), Digitally Controlled Oscillator (DCO), Clock Generation, Delta-Sigma Modulator.

I. INTRODUCTION

A phase-locked loop (PLL) is a closed-loop feedback system that controls the output signal of an oscillator to precisely track the phase and frequency of a reference signal. In modern systems-on-chip (SoCs), PLLs are an important functional block used for system clock generation, synchronization, and frequency synthesis [1].

While PLLs are traditionally fully analog or mixed-signal blocks, modern SoCs are designed in deeply scaled process technologies, which are increasingly unfriendly towards analog design. In these technologies, conventional mixed-signal circuits are challenged by low voltage headroom, wide variation over process, voltage, and temperature (PVT), increased capacitor leakage, and limited layout possibilities due to strict design rules [2]. Additionally, when co-implemented with large digital designs, analog circuits are subjected to the large switching transients and interference inherent in the operation of digital circuits, further limiting performance. As a result, mixed-signal PLL design in these environments is increasingly complex, time-intensive, and process-specific, precluding easy portability between designs and process technologies as is often required between SoC generations.

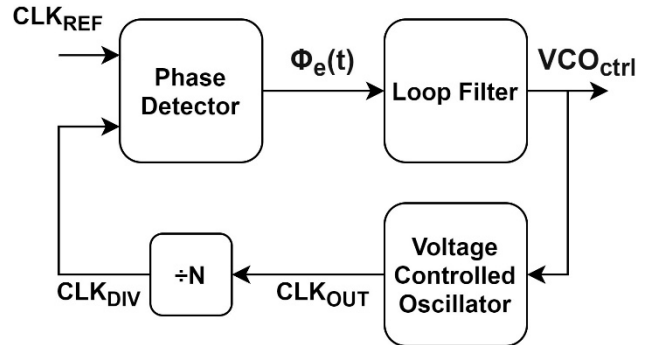


Fig. 1. General PLL structure.

On the other hand, all-digital phase-locked loops (ADPLLs) [2-8] — in which the inputs and outputs of all internal circuits are digital — have been gaining traction due to their improved stability, portability, and programmability over mixed-signal counterparts. As functionally digital circuits, ADPLLs are tolerant of PVT variations and noise [2]. Additionally, various works have demonstrated designs that are fully synthesizable with standard cells and are compatible with automated place and route (P&R) algorithms [4, 5, 7], allowing for integration and portability across existing digital design flows. Finally, programmability can easily be built into digital designs, enabling fast locking through dynamic adjustment of loop operation [2, 6] and improved resolution and noise performance through the use of DCO control dithering [3, 9]. For these reasons, ADPLLs are a promising solution for modern SoC design.

The remaining sections of this paper will focus on the design of state-of-the-art ADPLLs for digital clock generation and describe a high-performance, low-power ADPLL design in the ASAP7 7 nm PDK that gives the designer the freedom to programmatically trade fine output frequency resolution for low jitter, and vice versa. Section II will briefly describe how the different functional blocks that compose a PLL can be implemented digitally before examining how state-of-the-art ADPLL designs address some of the challenges unique to clock generation in modern SoCs, such as achieving fast lock times for dynamic frequency scaling, ensuring full compatibility with standard digital design flows while maintaining performance, and combining large DCO tuning range with fine output frequency resolution. Sections III and IV will describe the implementation of the proposed ADPLL architecture and present measurements verifying system performance. Finally, Sections V and VI will critically compare the presented implementation with the state of the art and outline the necessary steps to improve the design and integrate it in a fully-automated digital design flow.

II. STATE-OF-THE-ART ADPLL DESIGN

A. Making a PLL “All-Digital”

A traditional PLL consists of three primary building blocks in feedback configuration: a phase detector (PD), loop filter (LF), and voltage-controlled oscillator (VCO) [1]. The PD measures the phase difference between an input reference signal and the VCO output, which is often divided down to enable frequency multiplication. This phase difference is then processed by the LF into a signal that controls the operating frequency of the VCO, typically through a charge pump circuit. The VCO frequency is adjusted until its output phase matches the reference. When the phase error is zero, the loop is locked, and the phase and frequency of the oscillator exactly match those of the reference.

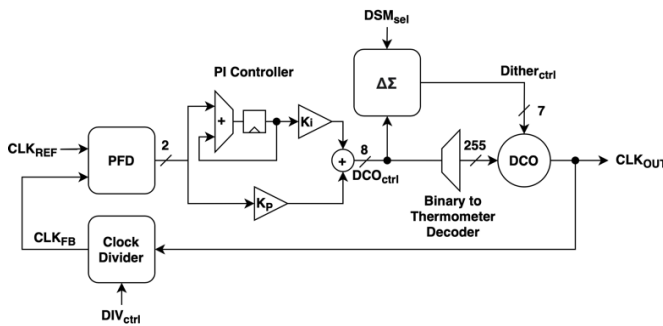


Fig. 2. Architecture of the proposed ADPLL.

As implied by its name, an all-digital PLL translates the analog components of a traditional PLL into their digital counterparts: the loop filter and charge pump become a digital loop filter (DLF), the voltage-controlled oscillator becomes a digitally-controlled oscillator (DCO), and any analog phase detection scheme is digitized, with phase frequency detectors (PFDs), bang-bang/Alexander phase detectors (BBPDs), and time-to-digital converters (TDCs) serving as preferred architectures.

B. Design Challenges and State-of-the-Art Solutions

For digital clock generation in modern SoCs, ADPLLs must be able provide output frequencies on the order of GHz, low peak-to-peak jitter and fine output frequency resolution to minimize clocking constraints, wide frequency tuning range and fast lock time compatible with dynamic voltage and frequency scaling (DVFS) techniques, and, ideally, simple integration with automated P&R flows [4]. Higher output frequencies come about naturally in more deeply scaled processes, so this is an area in which ADPLL designers do not have to expend much effort, but the other listed needs present unique design challenges.

In general, state-of-the-art designs minimize peak-to-peak jitter by making the coarse frequency steps of the DCO, as set by the DCO controller (the DCO gain), as small as possible, thereby reducing the quantization error that leads to deterministic jitter at the output [3]. The DCO designs presented in [3] and [7] achieve this by subdividing the available DCO tuning range over hundreds of steps, yielding

relatively fine “coarse” control. For both designs, frequency tuning is implemented through the enabling and disabling of individual delay elements within an interconnected ring structure composed of hundreds of delay elements. Enabling one delay element marginally increases the drive strength of its oscillator stage, and, for an approximately fixed capacitive load set by the base structure, this results in a slightly increased output frequency. However, this type of design quickly becomes complex and unfriendly to any automatic P&R program, as any irregularities in routing can greatly alter the DCO transfer characteristic, potentially threatening to render it useless by breaking its monotonicity. To address this concern, embedded calibration schemes have been proposed to inform the selections of the DCO controller according to the reality of the structure post-P&R, increasing design complexity to ensure robust operation [4, 7].

To achieve a wide frequency tuning range while avoiding the need for complex routing and maintaining fine output frequency resolution, coarser frequency tuning, such as through control of entire DCO rows rather than individual elements can be combined with a dithering mechanism which pseudo-randomly fine tunes the output frequency. This dithering, combined with the feedback action of the loop, acts to effectively improve the DCO resolution, and is typically implemented through delta-sigma modulation of the LSBs of the DCO control word [3, 9]. The implementation of dithering has the added benefit of reducing spurious tones in the output, which improves system-level performance when the generated clock is used for communication applications.

Finally, the lock time of the ADPLL is set by the bandwidth of the loop, which can be controlled primarily through the DLF. Fast locking requires a high instantaneous bandwidth, and can be achieved through dynamic loop bandwidth adjustment, implemented practically within a proportional-integral (PI) controller DLF by initially using a high integral gain for fast locking before transitioning to a lower integral gain to minimize dynamic and steady-state error [2]. To further speed up the locking, a frequency detection block can be introduced to quickly calculate the target code for the DCO [8].

III. PROPOSED ADPLL ARCHITECTURE

A. System Overview

Fig 2. shows the proposed ADPLL architecture, which consists of a PFD, a PI controller, a configurable-order delta-sigma modulator (DSM), a ring-oscillator-based DCO, and a controllable clock divider. The PFD is implemented using the well-known D-flip-flop-based design found in [1] and compares the phase error between the clock reference and the divided DCO clock, outputting digital UP/DOWN signals conveying the necessary frequency change. In this work, the reference frequency is 500 MHz and a counter-based divider with configurable divisors of 2, 4, 6, and 8 is implemented. The PFD signals are then digitally filtered and interpreted into a suitable DCO control word by the PI controller. The integral path is implemented as a 15-bit UP/DOWN counter, while the proportional path is simply a

single-bit UP/~DOWN signal. K_I and K_P are set to 2^{-7} and 2^0 , respectively, which, when accounting for a DCO gain of 27.25 MHz/LSB, results in a phase margin of 90° and a unity gain bandwidth of 1.4 GHz as found using the expressions in [2]. The output control word is then fed to the DCO and DSM, which will be explained in detail in the following sections.

B. DCO Design

The DCO topology, as shown in Fig 3. is based heavily on the designs proposed by [3] and [7]. The design is composed of 255 controllable and 16 baseline (always-on) interconnected three-stage ring oscillator rows, with C2MOS inverters acting as the basic inverting delay cells within these rows. Nine additional cells are allocated across three rows for

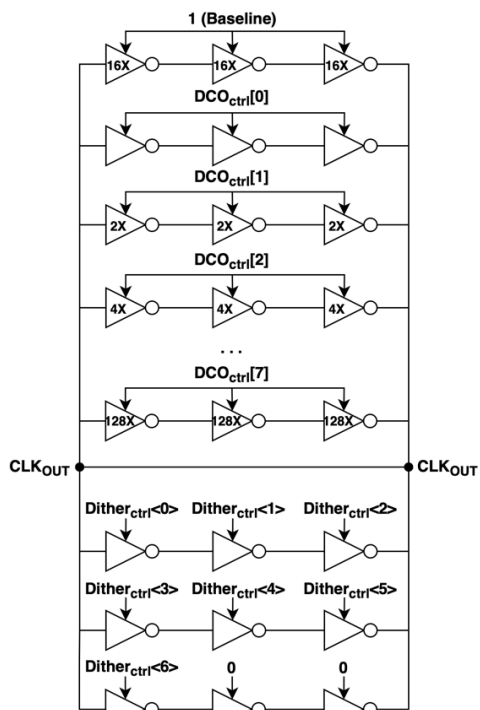


Fig. 3. Implemented DCO topology. The shown binary control encoding would be replaced by thermometer encoding in practice.

dithering, with two turned off permanently and seven selectable by the dithering outputs of the DSM. Coarse tuning of the output frequency is achieved by directly enabling rows according to the digital control word, interpreted as an unsigned binary number. By increasing the number of rows that are turned on, the drive strength of each oscillator stage increases proportionally, for an approximately fixed capacitive load. As a result, the output frequency transfer characteristic is theoretically very close to linear. In practice, to minimize entropy in row selection and ensure predictable frequency steps, the binary control word would be translated to a thermometer encoding, in which adjacent rows are only enabled or disabled in sequence, with output transfer function characterization done on the DCO after layout. However, due to the time constraints associated with this project and PDK model inaccuracies, these final steps were not pursued. Finally, linear fine tuning of the output frequency for

dithering is implemented through the enabling of individual C2MOS cells in the three allocated dithering rows, with each fine frequency step therefore corresponding to approximately one third of a coarse LSB.

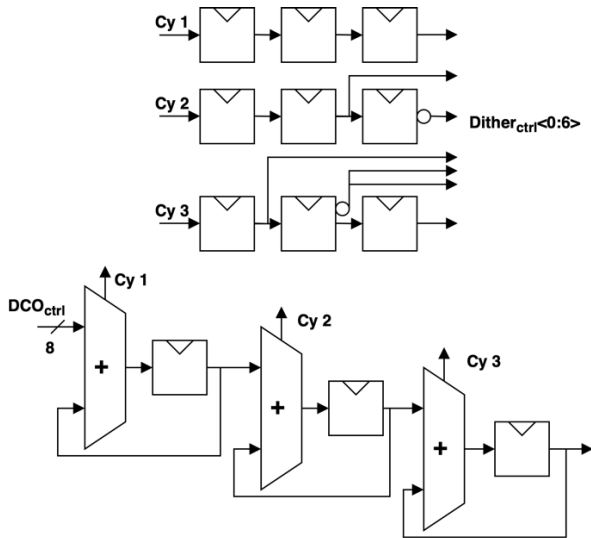


Fig. 4. Implemented third-order DSM. Adapted from [3].

C. DSM Implementation

The DSM is used to increase the resolution of the DCO by generating a dithering signal to pseudo-randomly control the fine resolution DCO control signals. The implemented DSM is based on a third-order MASH architecture similar to that found in [3]. As shown in the system-block diagram in Fig. 4, the DSM operates on the entire 8-bit control word, which is fed through a cascade of 3 accumulators, with each clocked at the reference frequency. The carry out signals of each of the accumulators are then combined through a delay block to produce a 7-bit dithering control word according to the equation,

$$O_{\Sigma\Delta} = C_1 \cdot D^3 + C_2 \cdot (D^2 - D^3) + C_3 \cdot (D - 2D^2 + D^3) \quad (1)$$

where $D \equiv z^{-1}$ is the delay operator [9].

Each of the accumulators can be clock gated such that a second-order and first-order DSM can be used instead, where the order corresponds to the number of active accumulators. Alternatively, the entire DSM can be clock gated to remove it completely from the control loop.

IV. SIMULATION AND MEASUREMENTS

A. Hardware Implementation and Test Bench

The proposed ADPLL was implemented in ASAP7, a predictive 7 nm FinFET CMOS technology. Given the analog operating nature and relatively higher design complexity of the DCO compared to the rest of the system, a two-pronged design and simulation approach was adopted, using transistor (SPICE) level design and simulation for the DCO and logic-synthesis and delay-annotated logic simulation for the remaining system blocks.

The DCO was implemented at the schematic level in Cadence Virtuoso, referencing ASAP7 transistor SPICE models, and frequency tuning, supply response, and power information was extracted through an automated Spectre test bench controlled via an OCEAN script. 150 fF of wiring capacitance was added to each node in the top-level design to better match simulations to realistic operating conditions. DCO power for each supply voltage was estimated as the average power consumed by the block at the highest possible frequency, with all delay cells enabled. The maximum power consumption at nominal operating conditions was found to be 2.2 mW.

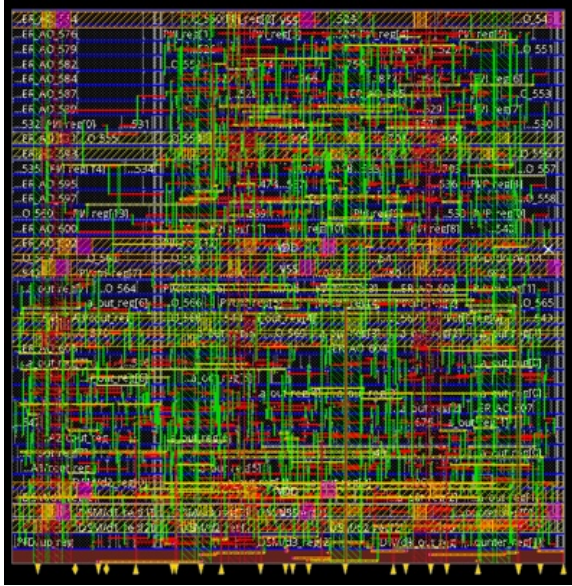


Fig. 5. ADPLL floor plan in ASAP7 predictive technology ($40\mu\text{m} \times 40\mu\text{m}$). The DCO is excluded.

The remaining system-level blocks (PFD, PI controller, DSM, and divider) were implemented in Verilog and synthesized using standard EDA tools. In order to avoid highly complex and time-consuming transistor-level simulations, a Verilog model of the DCO was created using a look-up table (LUT). In this LUT, DCO output frequency was specified as a function of the DCO and DSM control words in accordance with the Spectre simulations of DCO operation described above. By integrating this model with the synthesizable HDL implementations of the other blocks, the full system could be simulated efficiently in Synopsys VCS.

Fig. 5 shows the generated floor plan of the ADPLL on a $40\mu\text{m} \times 40\mu\text{m}$ ($1600\mu\text{m}^2$) die, excluding the DCO. The area of the synthesizable, system-level blocks was determined to be $932\mu\text{m}^2$, with the total DCO area estimated at $84\mu\text{m}^2$ based on the ASAP7 C2MOS layout outlined in [10], indicating that the full system can easily fit within the allocated die size. The power consumption of the system blocks excluding the DCO was determined using static power analysis with Cadence Voltus to be $96\mu\text{W}$. Therefore, the power, as expected, is dominated by the DCO and is estimated to be 2.3 mW at nominal operating conditions.

B. DCO Characterization

As indicated in Figs. 6 and 7, the simulated coarse and fine frequency tuning ranges at nominal operating conditions show that both transfer characteristics are highly linear, with the fine tuning range almost exactly matching a linear fit. In addition, coarse transfer characteristics were simulated for supply voltages (and logic high levels) down to 0.3 V, with the results presented in Fig. 8 and summarized in Table I. These data show that the same DCO design can be applied to a broad range of target output frequencies through scaling of its supply voltage, with linearity retained across such changes. Furthermore, this scheme promises the ability to conserve power by decreasing the DCO supply voltage, allowing the system-level designer to select the operating conditions that best match the needs of their specific application (e.g., low or high power and frequency). Simulations of fine frequency tuning and system-level performance were not conducted for the lower supply voltages, although fundamental system behavior is not expected to change drastically in response to different operating conditions.

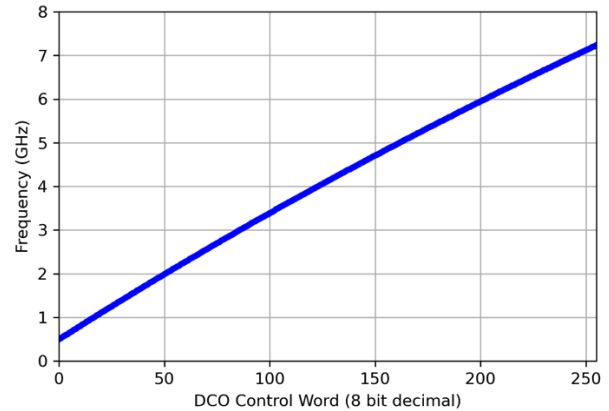


Fig. 6. Simulated tuning curve for DCO with nominal supply, TT devices, and at 27°C . The LSB is 27.25 MHz.

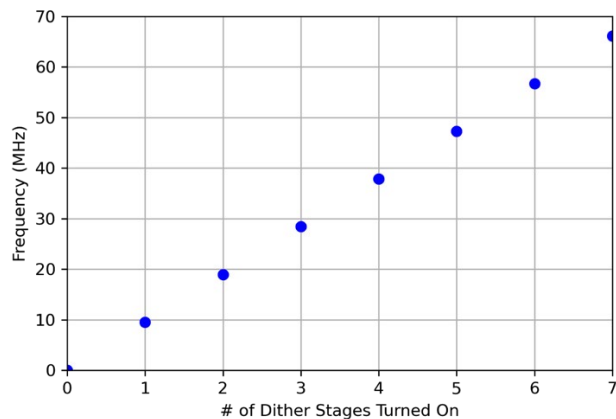


Fig. 7. Simulated fine tuning range for dithering. The Y axis represents frequency change relative to the nominal DCO operating frequency (when all dithering stages are off) at a given control word when X dithering stages are turned on. The frequency step between dithering levels is 9.5 MHz.

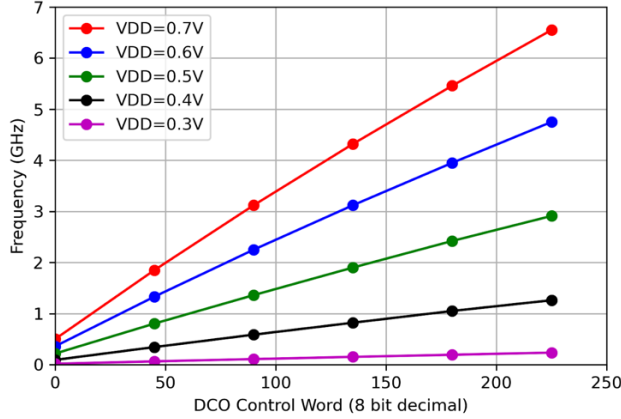


Fig. 8. Simulated DCO tuning curves at different supply voltages demonstrating available design tradeoff between power and DCO tuning range.

TABLE I
DCO OPERATION AT DIFFERENT SUPPLIES

Power Supply (V)	0.7	0.6	0.5	0.4	0.3
DCO Power (mW)	2.2	1.16	0.485	0.131	0.013
Output Frequency (GHz)	4	3	1.5	0.800	0.150
DCO Power/Freq. (mW/GHz)	0.55	0.387	0.323	0.164	0.087
Max Frequency (GHz)	7.24	5.3	3.25	1.41	0.263

C. System-Level PLL Characterization

As identified earlier in this report, two key PLL performance metrics for clock generation applications are peak-to-peak jitter, which helps determines maximum clocking frequency, and lock time, which is important for dynamic voltage frequency scaling (DVFS) applications where the PLL must quickly adjust output frequency to account for controlled changes in power supply voltage.

Fig. 9 shows the dynamic operation of the ADPLL with a second order DSM during initial lock and then in response to a frequency change. To simulate a DVFS scenario, the divider ratio is lowered such that the output frequency is changed from 4 GHz to 3 GHz and back again. The lock time is found to be 9 μ s, corresponding to 4500 reference cycles. The locking is relatively slow due to the fact that a very small integral gain and, thus, small loop bandwidth was chosen to provide high frequency stability following lock. Consequently, when the PLL is out of lock, the integral path responds slowly to the phase error, resulting in a lengthy lock time. To improve locking speed, a variable gain PI controller such as that in [2] could be implemented. However, due to time constraints such a system was not employed.

Since ASAP7 lacks sufficient noise models, only measurements of deterministic jitter, which results from the finite resolution of the DCO, were feasible. These measurements were made by simulating the PLL in lock and analyzing the systematic variation in the clock period with a

constant reference of 500 MHz and a divider ratio of 8. Peak-to-peak and rms jitter were calculated over 30,000 clock cycles, which is well over the 2^{10} cycle threshold reported in [3] for accurate measurements.

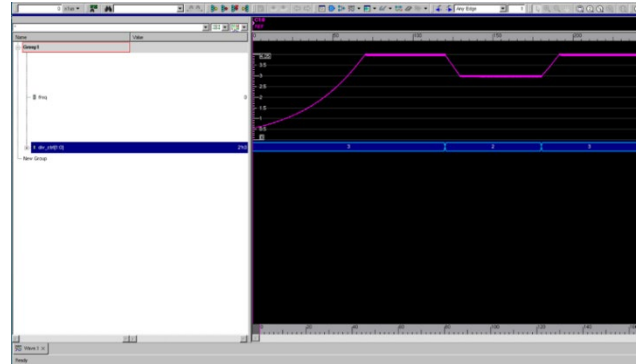


Fig. 9. Simulation of dynamic PLL operation with time scale in μ s. The divider ratio is initially set to 8 such that the PLL initially locks at 4 GHz. The divider ratio is then changed to 6 (3 GHz) and back to measure lock time.

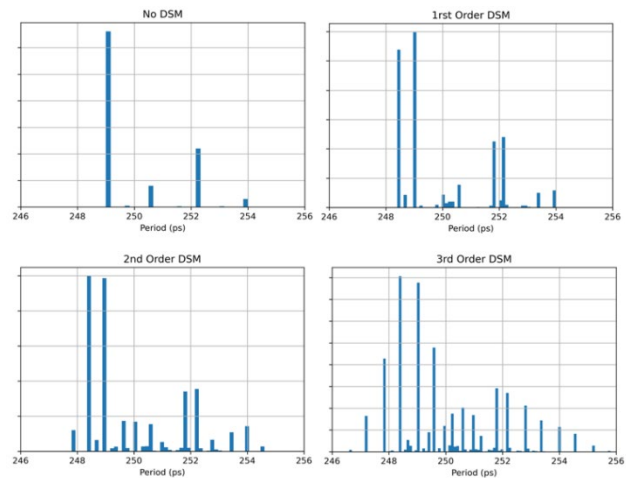


Fig. 10. Period histograms of PLL clock output in lock condition over increasing DSM order. The reference frequency is 500 MHz and the divider ratio is set to 8, resulting in nominal output frequency of 4 GHz and a period of 250 ps.

To understand the effect of changing the order of the DSM, ADPLL operation in conjunction with each of the selectable DSM orders (0, meaning that the DSM was disabled, 1, 2, and 3) was observed. The measured rms jitter and peak-to-peak jitter values for each of these experiments are reported in Table II. As observed in these data the peak-to-peak jitter represents less than 4% of the nominal clock period of 250 ps and the rms jitter is less than 1%. To fully understand the effect of changing the DSM order, period histograms for each DSM are shown in Fig. 10. Here, it is seen that for higher order DSMs, the effective DCO resolution is increased through the introduction of intermediary dithering levels between coarse DCO LSBs. As observed in the reported measurements, this results in a finer frequency resolution, but slightly increased jitter for higher-order DSMs due to the spectral spreading induced by the dithering. It should also be noted that the full dithering range, which is utilized with the third-order DSM, extends over more than 2

LSBs, and, thus, provides negligible frequency resolution enhancement over the 2nd-order implementation.

TABLE II
MEASURED JITTER VS. DSM ORDER

DSM Order	Pk-to-Pk Jitter (ps)	RMS Jitter (ps)
0	5.00	1.54
1	5.60	1.70
2	6.80	1.76
3	9.20	1.87

V. DISCUSSION

A comparison of this work with prior works is presented in Table III. As mentioned previously, it should be noted that the reported jitter measurements in this work only include deterministic effects and do not account for the random phase noise of the system. Noting this and the simulated nature of our results compared to those of the reference works, it is observed that our ADPLL model achieves a DCO tuning range and jitter performance comparable to that of the prior art. Due to the relative simplicity of the presented design and use of a more advanced technology node, the predicted power consumption and area figures are significantly less than those reported in other works. It is also observed that the reported lock time is greater than that in [2] due to the absence of a dynamically-adjustable PI controller or similar fast-locking mechanism

TABLE III
COMPARISON TO PRIOR WORKS

	[4]	[3]	[2]	This work*
Process	14 nm	65 nm	0.18 μ m	7 nm
Tuning Range (GHz)	1.0-5.5	0.5-8	0.25-1.367	0.5-7.239
RMS Period Jitter (ps)	1.29	0.7 @4GHz	8.884 @1.25GHz	1.76 @4GHz
PK-Pk Jitter (ps)	-	-	32.5 @1.25GHz	6.80 @4GHz
Lock Time (μ s)	-	-	2.9	9
Power (mW)	9.7	15.6 @4GHz	35 @1.25GHz	2.3 @7.24GHz
Area (mm ²)	0.009	0.03	0.7735	0.0016

*Reported results are for the architecture using the second-order DSM and nominal supply.

VI. CONCLUSION

This paper describes the challenges associated with designing ADPLLs for clock generation in modern SoCs, discusses state-of-the-art solutions to these challenges, and presents a representative ADPLL design for this application space. Implementation of the presented ADPLL architecture

in a nm-scale CMOS technology is validated through full system design simulation using the ASAP7 7 nm predictive PDK. Detailed characterization and simulated measurements of the DCO and PLL show that a wide tuning range (0.499-7.239 GHz), relatively fine frequency resolution (~9 MHz), and a low peak-peak jitter of 6.8 ps compatible with clock generation and DVFS can be achieved simultaneously using this architecture. Furthermore, the trade-offs between power consumption and tuning frequency range in DCO design and the advantages of increasing DSM order are elucidated.

In order to improve the applicability of this design to DVFS applications, the lock time should be improved, possibly through the use of a fast-locking algorithm similar to that shown in [2]. Additionally, this design falls just short of being fully compatible with standard digital EDA flows and further research is required to integrate the proposed DCO architecture within the design while mitigating performance degradations due to automated P&R. As mentioned in [4, 7], a practical but complex solution would involve the addition of a built-in automatic calibration scheme to obviate the need for manual layout or calibration.

REFERENCES

- [1] R. E. Best, "Phase-locked loops: Design, simulation, and applications," In *Phase-locked loops: Design, simulation, and applications* (6th ed.). New York, NY: McGraw-Hill, 2007.
- [2] J. Lin and C. Yang, "A Fast-Locking All-Digital Phase-Locked Loop With Dynamic Loop Bandwidth Adjustment," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2411-2422, Oct. 2015.
- [3] J. A. Tierno, A. V. Rylyakov and D. J. Friedman, "A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI," in *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 42-51, Jan. 2008.
- [4] D. M. Moore, T. Xanthopoulos, S. Meninger and D. D. Wentzloff, "A 0.009 mm² Wide-Tuning Range Automatically Placed-and-Routed ADPLL in 14-nm FinFET CMOS," in *IEEE Solid-State Circuits Letters*, vol. 1, no. 3, pp. 74-77, Mar. 2018.
- [5] W. Deng et al., "A Fully Synthesizable All-Digital PLL With Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique," in *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 68-80, Jan. 2015.
- [6] Y. Ho and C. Yao, "A Low-Jitter Fast-Locked All-Digital Phase-Locked Loop With Phase-Frequency-Error Compensation," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 5, pp. 1984-1992, May 2016.
- [7] Y. Park and D. D. Wentzloff, "An all-digital PLL synthesized from a digital standard cell library in 65nm CMOS," 2011 IEEE CICC, San Jose, CA, USA, 2011, pp. 1-4.
- [8] C. Chung and C. Lee, "An all-digital phase-locked loop for high-speed clock generation," in *IEEE JSSC*, vol. 38, no. 2, pp. 347-351, Feb. 2003.
- [9] 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.
- [10] D. Fritchman and W. Rahman, "Adaptive Clocking Techniques for SoC Supply Droop Response in Predictive 7nm CMOS," UC Berkeley EE 241B Course Project, May 2020.