Analysis and Design of All-digital Phase-locked Loops for SoC Integration in Deeply Scaled Technologies

Averal Kandala[1] and Micah Roschelle[1]
[1]Department 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 perform critical operational functions in SoCs, such as clock generation and frequency synthesis. 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 designs for their stability over PVT variations, compatibility with automated digital design flows, portability between technologies, and dynamic programmability. All-digital phase-locked loops (ADPLLs) consist of a digital phase detector, digital loop filter (DLF), and digitally controlled oscillator (DCO). This work will compare and contrast various state-of-the-art designs for each of these functional blocks and determine an optimal topology for integration with an automated digital design flow in a nm-scale technology. A plan will then be laid out to evaluate a number of described topologies in the ASAP7 7 nm predictive PDK through a series of ADPLL schematic testbenches assessing lock range, DCO linearity, jitter, and power.**

Index Terms – All-Digital Phase-Locked Loop (ADPLL), Digitally Controlled Oscillator (DCO), Phase Frequency Detector (PFD), Digital Loop Filter (DLF), SoC Design.

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), especially those that integrate radio frequency (RF) functions with digital signal processing, PLLs are an important functional block used for system clock generation and synchronization, frequency synthesis, and RF modulation and demodulation [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 noise 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.

On the other hand, all-digital phase-locked loops (ADPLLs) [2-13]—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 [3]. Additionally, various works have demonstrated designs that are completely implemented in a hardware description language (HDL) [4, 5], fully synthesizable [5, 6] with standard cells [7, 8, 10], and are compatible with automated place and route (P&R) algorithms [7, 8], allowing for integration and portability across existing digital design flows. Finally, programmability can easily be built into digital designs, enabling high performance through dynamic adjustment of loop operation [2, 9, 10]. For these reasons, ADPLLs are a promising solution for modern SoC design.

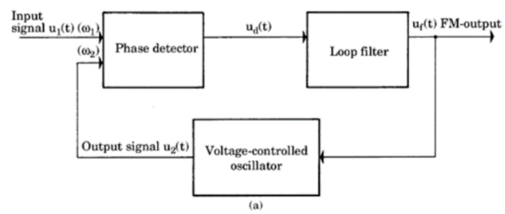

Fig. 1: Elementary system block diagram of a PLL [1].

The remaining sections of this paper will focus on the design and implementation of state-of-the-art ADPLLs. Section II will describe the basic block diagram of a PLL before providing an in-depth analysis of the common digital implementations of each functional block found in state-of the art ADPLLs. Section III will lay out an implementation plan and experimental setup for a proposed design in the 7nm ASAP7 process. Finally, Section IV will connect the proposed implementation with our original motivations.

II. State-of-the-Art ADPLL Design

A. Basic PLL Architecture and Operation

As shown in Fig. 1, 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 output of VCO. This phase difference is then processed by the LF into a signal that controls the operating frequency of the VCO. In the presence of insufficient input signal, the loop is open and the VCO oscillates at its free running frequency. As an input signal is acquired, the PLL transitions to capture mode, in which 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.

According to R. E. Best's classification, an ADPLL is a PLL that is not only exclusively built with logic devices, but in which all of its internal signals (phase error, frequency control word, etc.) are digital signals [1]. Thus, the traditional PLL blocks are replaced by a digital phase detector (DPD), digital loop filter (DLF), and digitally controlled oscillator (DCO). The following sections will delve into state-of-the-art digital implementations and architectures for each of these functional blocks.

B. Digital Phase Detectors

Digital PD implementations in state-of-the-art ADPLLs generally take one of two standard forms: phase frequency detector (PFD) based designs [2, 5, 9, 10] or time-to-digital converter (TDC) based designs [3, 7, 11].

Fig. 2 shows the implementation of a standard PFD, which consists of two D-flip-flops clocked separately by the reference and DCO output, as well as a NAND gate which takes the flip-flop outputs as inputs and develops the reset signal. If the rising edge of the reference signal is seen first by the detector, the UP signal goes high until the rising edge of the DCO output is detected and both flip-flops are reset. In this case, the frequency of the DCO must be increased to “catch up” with the reference. In the other case, when the DCO edge is seen first, the DN (“down”) signal is set high, signifying that a decrease in DCO frequency is necessary. Thus, the output signals of the detector (UP and DN) convey both the phase difference and the polarity of the required frequency change. The ability to detect phase and frequency simultaneously has been shown to lead to a fast locking condition, as when the input reference is faster than the DCO, a majority of UP signals are generated (the opposite is true for a slower reference) [1]. While the advantage of this design is its simple digital implementation and fast locking, one disadvantage is fixed phase error due to the “dead zone” of the detector. If the NAND delay is faster than the delay of the subsequent control circuitry, it is possible that no phase error will be detected. To minimize this dead zone, a digital pulse amplifier circuit (Fig. 2) implemented with cascaded AND gates has been proposed to amplify small phase error signals to a detectable level [10].

Another considerable drawback with PFD-based designs is that only the edges of the UP and DN signals are captured by the control path in all-digital loops and, thus, phase magnitude information is lost. Therefore, to achieve finer phase error resolution, TDC-based PDs have been proposed. An ADPLL with a TDC-based phase detector is shown in Fig. 3 [7]. The TDC takes the DCO output and reference signal as inputs, measures the time interval between the rising edges of

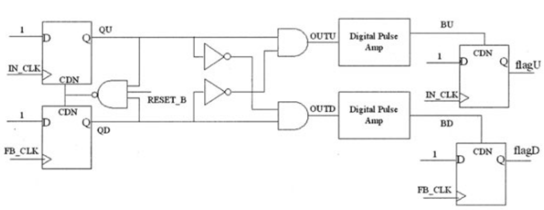

Fig 2: PFD implementation in [10]. The leftmost flip-flops and NAND gate implement a standard PFD, which produces UP (QU) and DN (QD) signals. An additional pulse amplifier stage implemented with cascaded AND gates improves detector dead zone.

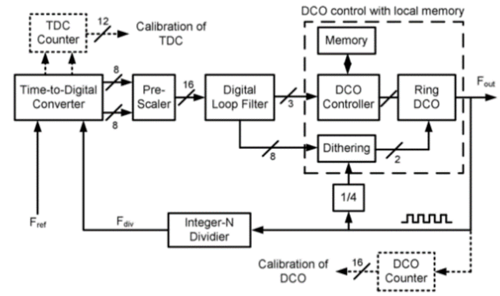

Fig. 3: A proposed ADPLL using a TDC as a phase detector [7]. The TDC is implemented with a DCO-based Vernier topology. To compensate for the buffer systematic offset in the DCOs due to automated P&R, a calibration unit is used to program the order of the in the DCO to allow for suitable adjustment during capture and lock phases.

the signals, and outputs a digital value representing the magnitude of the measured phase error. This block can be implemented with a chain of fixed-delay inverters [11], by using a Vernier topology with a counter and a DCO [7], or with equally spaced tap clocks and a flip-flop array [3]. While TDCs can provide extremely fine phase resolution, they do so at the cost of increased circuit complexity and sensitivity to PVT variations [2], complicating compatibility with automated P&R flows. To address these issues, PVT-insensitive topologies and post-fabrication calibration procedures have been proposed to correct for variation incurred by automated P&R [7].

In addition to standard PFD- or TDC-based topologies, other phase detection schemes have been proposed which use a Hilbert transformer [4] or injection-locking techniques [8]. While these designs have specific advantages, they again come at the cost of increased design complexity.

C. Digital Loop Filters

The DLF plays two important roles in ADPLL operation: first, it translates the output of the DPD into a value suitable for DCO control and, second, it determines the overall loop transfer function, which has significant implications for loop stability, noise tolerance, and lock time [1]. As a result, DLF topology selection is highly dependent on the selected DPD and the desired loop characteristics.

A DLF that is commonly used in PFD-based systems is the UP/DOWN counter [1]. In this design, a pulse-forming circuit translates incoming UP and DN pulses into a clock signal and a direction ($\overline{\mathrm{UP}}/\mathrm{DN}$) signal, which trigger a bidirectional UP/DOWN counter [1]. On the clock’s rising edge, the counter increments its N-bit output word by 1 if the direction signal is low and decrements the output word by 1 if the direction signal is high [1]. This system’s action can be approximated as that of a continuous-time integrator [1]; however, because the UP and DN pulses do not contain any information on the magnitude of the phase error, this DLF does not lend itself to rapid locking as a true integrator would. An extension of this design is the K counter, which requires the same clock and direction signals as the UP/DOWN counter but uses the direction signal to select which of two separate UP and DN counters is triggered by the clock. Furthermore, an additional loop parameter, K, is introduced

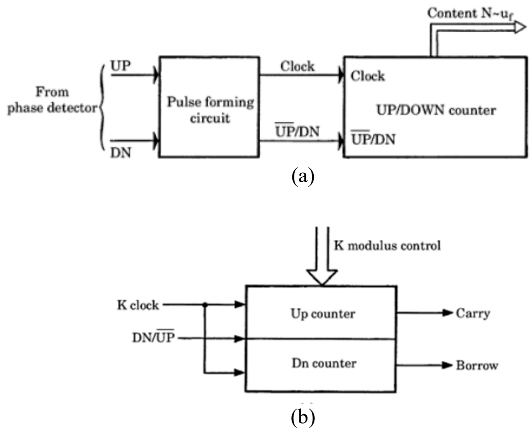

Fig. 4: Block diagrams of UP/DOWN counter (a) and K counter loop filters [1].

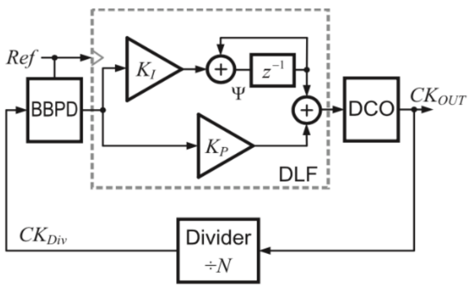

Fig. 5: Block diagram of a proposed ADPLL with a second order loop filter [2]. The integral and proportional gains, K_I and K_P, are controlled to dynamically adjust filter response, enabling fast locking (high bandwidth) during capture mode and high noise rejection (low bandwidth) in lock mode.

that represents the modulus of the counters such that, when the counters reach K – 1, they are reset to 0 on the subsequent edge. The output Carry and Borrow signals, shown in Fig. 4(b), represent the MSB of each counter and, thus, go high at K/2, modulating the frequency of the DCO [1]. This filter, therefore, acts as a simple average function and can be thought of as an approximate low pass filter with a single pole in its transfer function. As can be shown in the control theory analysis of PLLs, such a loop filter often results in a second order loop, in which the lock range is proportional to the bandwidth [1], limiting options if a narrow bandwidth, but large lock-range is desired.

To provide further design flexibility and improved performance, state-of-the-art PLLs use z-domain digital filters or proportional-integral-derivative (PID) controllers to synthesize higher order LFs [4, 5, 7, 9, 11]. A representative design is shown in Fig. 5 [2]. One advantage of such digitally implemented filters is that they are easily programmed and can be adjusted dynamically, allowing for the use of different filters in different operational modes. A significant result from PLL control theory is that a wide bandwidth allows for fast locking during capture mode, while a narrow bandwidth is necessary for limiting spurious tones (“spurs”) and noise during lock mode [2]. As a result, DLFs with dynamic loop bandwidth adjustment have been proposed to significantly improve lock time while maintaining performance during lock [2, 9].

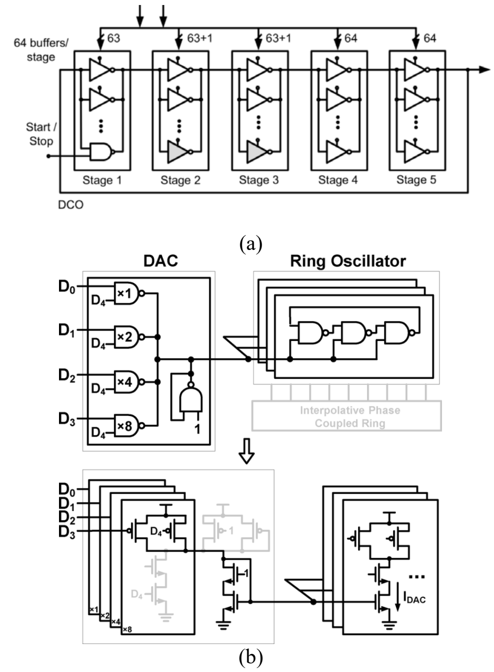

Fig. 6: Buffer selection (a) and current-output DAC (b) ring DCOs [7,8].

D. Digitally Controlled Oscillators

The DCO is the core component of any ADPLL, setting the output signal and closing the feedback loop. Many so-called “DCO” architectures exist which use digital-to-analog converters (DACs) to interface directly with a VCO [12]; however, this kind of architecture generally requires a relatively significant amount of analog/mixed-signal design and is therefore not typically compatible with an automated digital design flow. Among fully synthesizable designs that implement the DCO using HDL code or standard cells, the prevailing strategy involves the digital tuning of the frequency of a single ring oscillator (RO) or a ladder of interconnected ROs [5, 7, 8, 13, 14].

For the simplest such designs, the basic methodology used to digitally tune the frequency of the RO structure centers around the modulation of the drive strength of each stage of the structure via the input digital control word [7, 8, 13, 14]. Since the capacitive fanout of each stage is mostly set by the fixed physical topology of the DCO, this approach of changing the drive strength works to directly increase the frequency of oscillation based on the number of engaged delay elements.

Fig. 6(a) depicts a strategy for achieving this effect by placing a number of ROs in interconnected rows, with digital control over which specific delay elements to activate [7]. A similar approach is taken in [13]. In both implementations, “dithering” is included in the control scheme to increase frequency resolution and reduce spurs, which can arise due to the quantized nature of the DCO frequency [13]. The main advantage of this type of design is that, because every block is synthesizable, the entire DCO and its controller can be implemented through automatic P&R, and an attached calibration apparatus can be introduced to address systematic mismatch from the automated layout process [7]. However, a simpler design, involving row-based control of the RO ladder without any calibration or dithering mechanism, could suffice to provide basic proof of this concept, with precision being reduced and spurs arising at the two frequency codes closest to the reference frequency as a natural result [7].

Indeed, the design presented in [14] essentially follows this approach. This structure implements row selection by introducing modified transmission gates at the inputs of each delay buffer, in which the paths to the pull-down and pull-up devices of the buffers are kept separate [14]. The main drawback of this implementation is that the individual delay elements for this structure cannot be synthesized from HDL code or standard cells, and so some custom design must go into assembling the basic buffer for this topology. Additionally, while this design promises high DCO linearity and low phase noise, its operation within a closed loop has not been presented, and so other performance metrics, including lock range and spur rejection, are not assessed [14]. In particular, the relative simplicity of the frequency selection scheme suggests that significant spurs can arise due to output frequency quantization, as discussed above.

Finally, a DCO topology proposed in [8] achieves RO drive strength control using a current-output DAC cleverly implemented exclusively using standard cell NAND gates. Presented in Fig. 6(b), this design uses one NAND gate as a current mirror to the pull-down device stacks of the RO array, varying the current driven through the array by changing the voltage across this mirror. The larger the digital input word, the lower this voltage falls, yielding lower RO drive strength and lowering the output frequency. As with the structure shown in Fig. 6(a), a calibration scheme is used to address systematic mismatch resulting from automatic P&R [7, 8].

III. Implementation and Comparison Framework

Given the inherently modular structure of the ADPLL and diversity in implementation options for each stage, it makes sense to constrain the majority of any design exploration to one stage to ensure that a functioning system can be constructed and meaningfully characterized in a reasonable timeframe. The natural candidate for the focus of the design exploration for this project is the DCO, as it is the most complex block to implement and is also the block that most constrains the “digital” nature of any ADPLL. Therefore, the following implementation and comparison plan in the ASAP7 7 nm predictive PDK is proposed:

1. The PFD and UP/DOWN counter phase detection and loop filter topologies, synthesized from HDL code and standard cells, will compose the “harness” of an initial ADPLL design, as they are relatively simple to implement and can interface with any one of the highlighted DCO structures directly through a digital control word.
2. Each of the RO-based DCO topologies introduced in this paper will be designed at the netlist/schematic level, using HDL code and standard cells wherever possible, and characterized in a SPICE testbench with regard to linearity and phase noise, before being inserted into the described ADPLL harness.
3. With frequency division added in the loopback path to ensure an output frequency on the order of 1 GHz, an assessment will be conducted in a SPICE testbench on the lock range, jitter, and power of each complete system. The DCOs will then be compared based on these metrics, as well as suitability for inclusion in an automated digital design flow.
4. If time permits, automated P&R can be run on any synthesized designs and the performance of the laid-out ADPLLs with respect to the described metrics can be contrasted with that of the corresponding netlist-level descriptions. Finally, more advanced phase detection and control schemes can also potentially be explored.

IV. Conclusion

In summary, this paper introduces the basic components of any PLL and identifies the key defining qualities that allow a truly all-digital PLL design to be implemented in an automated digital design flow. The hope of the authors is that the design methodology that is elaborated as part of the next phase of this project will lay a roadmap for integrated ADPLL and SoC design in deeply scaled process technologies.

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] W. Liu, W. Li, P. Ren, C. Lin, S. Zhang and Y. Wang, "A PVT Tolerant 10 to 500 MHz All-Digital Phase-Locked Loop With Coupled TDC and DCO," in *IEEE Journal of Solid-State Circuits*, vol. 45, no. 2, pp. 314-321, Feb. 2010.

[4] M. Kumm, H. Klingbeil and P. Zipf, "An FPGA-Based Linear All-Digital Phase-Locked Loop," in *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 9, pp. 2487-2497, Sept. 2010.

[5] S. Radhapuram, T. Yoshihara, T. Matsuoka, “Design and Emulation of All-Digital Phase-Locked Loop on FPGA,” in *Electronics* 2019, 8, 1307.

[6] I. Ali et al., "An Ultra-Low Power, Adaptive All-Digital Frequency-Locked Loop With Gain Estimation and Constant Current DCO," in IEEE Access, vol. 8, pp. 97215-97230, 2020.

[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] 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.

[9] 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.

[10] 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.

[11] R. B. Staszewski *et al*., "All-digital PLL and transmitter for mobile phones," in *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2469-2482, Dec. 2005.

[12] A. Singhal, C. Madhu and V. Kumar, "Designs of All Digital Phase Locked Loop," *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, Chandigarh, India, 2014, pp. 1-5.

[13] 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.

[14] A. Tomar, R. K. Pokharel, O. Nizhnik, H. Kanaya and K. Yoshida, "Design of 1.1 GHz Highly Linear Digitally-Controlled Ring Oscillator with Wide Tuning Range," 2007 IEEE International Workshop on Radio-Frequency Integration Technology, Singapore, 2007, pp. 82-85.