Design Techniques for a Multi-Phase Injection-Based Eight-Phase 17-GHz Clock Generator for Multi-Phase Wireline Receivers

Bob Zhou, *Graduate Student Member, IEEE*, and Borivoje Nikolić, *Fellow, IEEE*

***Abstract*— Clock generation for high-speed wireline receivers must provide multiple clock phases with high-resolution rotation. To address this, an 8-phase 17 GHz clock generation circuit with built-in 6b rotation is presented. Multi-phase injection is used to perform reference-side phase rotation to efficiently generate and rotate eight clock phases. The injection method is analyzed with a model to study the introduced nonlinearity, and the effect of the injection strength is discussed. Designed by using BAG3++ for layout-aware design optimization, the proposed circuit achieves 98 fs RMS jitter and a measured DNLpp and INLpp of 1.26 and 4.05 LSB respectively, while consuming 33 mW.**

***Index Terms*— Phase rotator, multi-phase clock generation, injection locked ring oscillator, multi-phase injection, phase interpolator.**

I. Introduction

NEXT-GENERATION AI workloads are performed using highly interconnected computing systems for distributed and resource-efficient computing. This interconnect network creates a bottleneck for performance, driving the demand for faster and more power-efficient high-speed wireline systems. PCI-SIG continues to double the per-lane data rate every three years (Fig. 1), with the latest PCIe Gen 7.0 standard aiming for 256 Gb/s per pin [1]. Ethernet data rates double at a similar rate, with the recent 2024 standard targeting 200 Gb/s, doubling in less than 3 years [2].

To meet these requirements, long-reach wireline receivers over 100 Gb/s predominantly use analog-to-digital converter-based (ADC-based) frontends and digital or DSP-based equalization. In recent publications, the successive-approximation-register (SAR) ADC (Fig. 2) has been the architecture of choice for its power and area efficiency in advanced process nodes and for its ability to scale to higher data rates due to multi-phase sampling or time-interleaving. Multi-phase sampling has existed in wireline receivers for decades in dual data rate (DDR) and quad data rate (QDR) mixed-signal architectures. While most published >100 Gbps receivers use a 4-way interleaved SAR ADC, newer receivers are pushing into 8-way interleaving [3], [4], [5], [6]. Recently published receivers above 200 Gbps have employed 8-way interleaving [7] and even 16-way interleaving [8], [9].

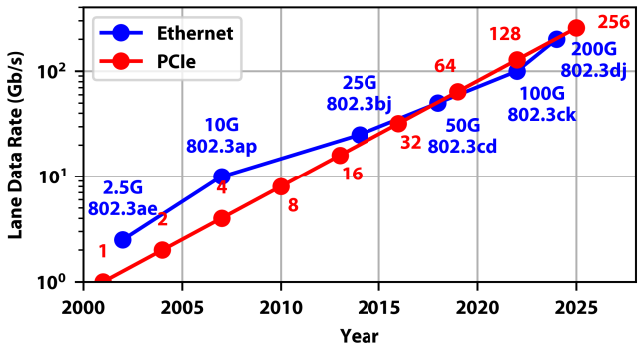


Fig. 1. I/O bandwidth (data rate per lane) of PCIe [1] and Ethernet [2] standards.

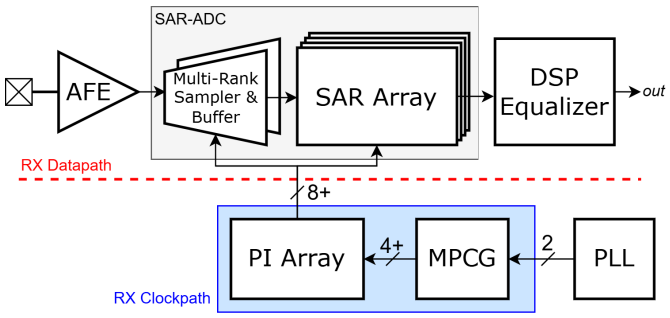


Fig. 2. Typical ADC-based wireline receiver for >100 Gbps PAM4. The proposed clock generator would replace the RX clock path (blue box).

The receiver-side clock generation is burdened to produce both high-frequency and multiple phases of clocks. These generated clocks must also have extremely low jitter since these receivers are used with high-loss channels where the receiver eye is significantly degraded. At 112 Gbps PAM4, the available unit interval (UI) is 17.8ps. To meet the Ethernet standard of 6-sigma 0.05UI peak-to-peak (P2P) target, less than 150 fs RMS jitter can be tolerated. Similarly, at 224 Gbps PAM4, less than 75 fs RMS jitter can be tolerated.

Finally, wireline receivers require a phase rotation mechanism to align the receiver sampling phase. Sampling phase alignment must be performed on all the generated phases and must have high resolution to compensate for the degraded eye. Though the required phase resolution is implementation-specific, the 0.05UI target can be borrowed from the jitter

Received 7 December 2024; revised 28 February 2025 and 10 April 2025; accepted 14 April 2025. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) Intel 22FFL and Packaging Technologies for Terahertz Radios (I-TERA) Program. This article was recommended by Associate Editor J. Craninckx. *(Corresponding author: Bob Zhou.)*

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94704 USA (e-mail: bob.linchuan@eecs.berkeley.edu).

Digital Object Identifier 10.1109/TCSI.2025.3563517

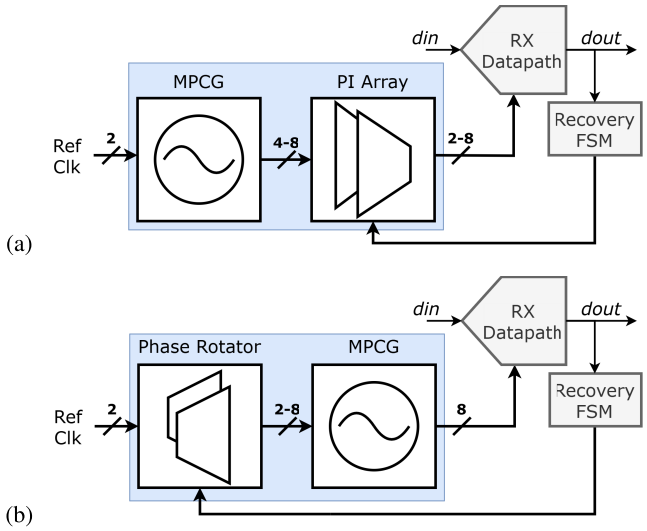


Fig. 3. (a) The typical clock generation system requires a PI to align each generated phase. As the number of clock phases increases, this significantly contributes to clock generation power. (b) An alternative approach is to apply rotation on the reference side.

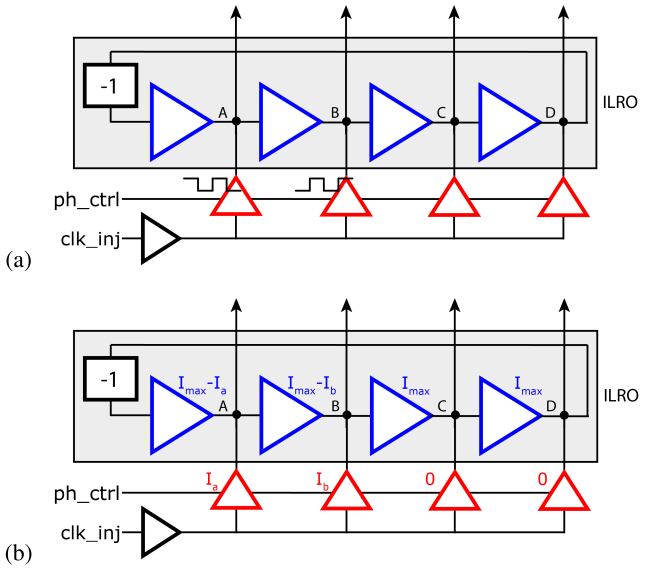


Fig. 4. Multi-phase injection-locked phase rotator concept. In this configuration, the injection phase will lock between nodes A and B. In (a), a fixed injector strength is used, and dithering is performed to achieve intermediate lock. In (b), the explicit strength of the injectors is controlled up to the total strength I_{max}.

specification. For this, at least 50 steps are required, giving a 6-bit resolution minimum. These collective challenges make clock generation an increasingly large part of the wireline power budget.

A typical wireline receiver clock generation system is shown in Fig. 3(a). A multi-phase clock generation (MPCG) circuit, such as an injection-locked ring oscillator (ILRO) or a phase-locked loop (PLL), is used to generate multiple phases. Then, a phase interpolator (PI) rotates the clock phase for sampling phase alignment. The number of PIs scales with the number of sampling phases, greatly adding to the power and area cost for these many-phase ADCs. To avoid this cost, phase rotation must be performed on the reference side of the MPCG instead of the output side, as shown in Fig. 3(b). This efficiently rotates all the generated phases at once.

In ILRO-based MPCGs, reference-side rotation can be accomplished by using multi-phase injection (MPI). MPI injects a single reference phase into one or more oscillator nodes, as conceptually illustrated in Fig. 4. The core ring oscillator has injectors in each stage. *ph_ctrl* is used to select which node receives injection. Selecting a single node, e.g., node A or B, will align the injection signal with that node. This efficiently rotates all the oscillator phases at once. MPI extends this idea by injecting into two adjacent stages simultaneously, e.g., nodes A and B. The injection signal will then align somewhere between nodes A and B, achieving intermediate steps beyond the number of nodes in the oscillator.

Prior work on MPI-based phase rotators has primarily focused on time-modulated solutions, as shown in Fig. 4(a) [10], [11]. These methods dither between nodes to achieve the intermediate steps. Though this is hardware efficient, it introduces undesired deterministic jitter. A static method is preferred, where the explicit strengths of the injectors are controlled, as in Fig. 4(b). Prior work on static methods has achieved only 3 bits of resolution and also does not meet the high frequency requirements of modern receivers [12].

This article presents an 8-phase 17 GHz clock generator with integrated 6b phase rotation in a 16 nm FinFET process [13]. This clock generator performs reference-side rotation using a static MPI technique to simultaneously and efficiently rotate all generated phases, as in Fig. 4(b). A system model is introduced to predict the static nonlinearity introduced by MPI with good correlation with measured results. The proposed clock generator layout is generated by using the Berkeley Analog Generator (BAG), allowing for rapid layout-aware design optimization in advanced process nodes.

This article is organized as follows. Section II describes the proposed phase rotation mechanism and modeling methods. Section III presents the circuit architecture and implementation details of the critical blocks, including the core oscillator, current DACs, and input and output buffers. Circuit generation and design flow using BAG3++ are discussed in Section IV. Finally, Section V summarizes the measurement results, and Section VI concludes the work.

II. Nonlinearity Modeling

The proposed static rotation method is shown in Fig. 4(b). The core circuit is a ring oscillator with injectors in each stage. Reference rotation is performed by changing which oscillator node locks to the reference, rotating all the oscillator's output phases together. The circuit can phase-lock the reference with node A by setting $I_a \neq 0$ and $I_b = 0$. Similarly, the circuit can phase-lock the reference with node B by setting $I_b = 0$ and $I_a \neq 0$. To achieve more steps than the number of nodes in the oscillator, MPI is used to set $I_a \neq 0$ and $I_b \neq 0$, phase-locking the reference between node A and B and giving us the desired intermediate steps. The phase lock can then be progressively shifted from node A to node B by gradually increasing I_b while decreasing I_a. This same procedure can be repeated for nodes B and C and so forth to achieve full rotation.

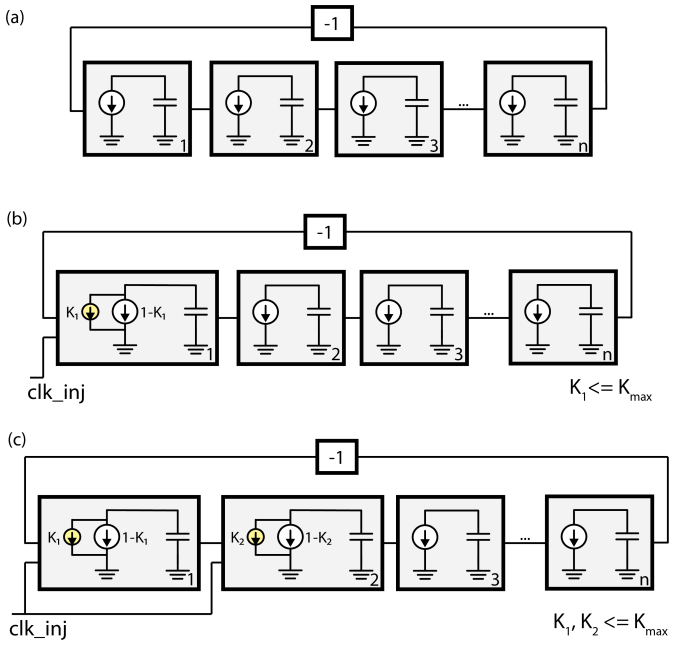


Fig. 5. Development of the multi-phase injection model. (a) General oscillator model. (b) Single-phase injection model. (c) Multi-phase injection model.

Theoretical analysis must be introduced to study the static rotator phase integral non-linearity (INL) introduced by this MPI scheme. Modeling methods for injection oscillators may be categorized into frequency domain, phase domain, and time domain models [14], [15]. Frequency or phasor domain analysis is well suited to study the locking range and jitter filtering of injection oscillators. Prior publications using these methods have focused on injection using multiple phases to extend locking range [16] rather than for phase rotation. Phase domain analysis typically requires an impulse sensitivity function, and so is less intuitive from a modeling perspective.

A time domain model was chosen for this study as it is well-suited for ring oscillators and is intuitive for understanding the phase interactions of each stage [17]. Prior modeling efforts in MPI-based phase rotators has focused on time-modulated methods [11]. As this work involves a static MPI method, a new model is needed. The proposed model is shown in Fig. 5. In this first-order case, only current sources and capacitors are considered. Rise and fall times are ignored.

A. Derivation With One Injection Stage

Fig. 5(a) starts with a generic oscillator with no injectors. A stage is defined as a current source charging a capacitor. When the voltage on the capacitor in a given stage exceeds a threshold value V_{th}, the subsequent stage's current source starts charging. This time from when a stage starts charging until it triggers the next stage is the propagation delay t_p. Assuming a current source turns on at $t = 0$ and has a fixed current I_d, t_p can be derived as follows.

$$I(t) = I_d \tag{1}$$

$$V_c(t_p) = V_{th} = \int_0^{t_p} \frac{I(t_p)}{C} dt \tag{2}$$

$$t_p = \frac{V_{th} C}{I_d} \tag{3}$$

Fig. 5(b) introduces a single injection stage to the model. This injection stage adds an auxiliary current source triggered by the external injection clock. The main current source is still triggered by the previous oscillator delay stage. The single current I_d is now split between the main and the auxiliary current sources, with a fraction K_1. K_1 may be controlled by the circuit up to a maximum value K_{max}. Assume the injector will turn on at some time $t = t_d$ and the main source at $t = 0$. The injection stage can be modeled by the following equations.

$$I_1(t) = \begin{cases} (1 - K_1) I_d & \text{if } t < t_d \\ I_d & \text{if } t \geq t_d \end{cases} \tag{4}$$

$$V_1(t_1) = V_{th} = \frac{1}{C} \begin{cases} (1 - K_1) I_d t_1 & \text{if } t_1 < t_d \\ I_d (t_1 - K_1 t_d) & \text{if } t_1 \geq t_d \end{cases} \tag{5}$$

$$t_1 = \begin{cases} \dfrac{t_p}{(1 - K_1)} & \text{if } t_1 < t_d \\ t_p + K_1 t_d & \text{if } t_1 \geq t_d \end{cases} \tag{6}$$

To determine the phase alignment between the oscillator and the injection clock, the free-running oscillator period is assumed to be the same as the injection period. Under injection lock, the total combined delay contributed by all stages should equal the injection clock period. To meet this requirement, the injection stage must contribute the same delay as any other non-injection stage, i.e., $t_1 = t_p$. For non-zero injection ($K_1 > 0$), this forces $t_d = 0$, meaning the injection clock phase will align with the oscillator phase at $t = 0$, as expected from a single-phase injection oscillator. This is illustrated in Fig. 6.

B. Derivation With Two Injection Stages

Fig. 5(c) introduces multi-phase injection to the model. Two adjacent stages now contain injector current sources, both triggered by the external injection clock. Each injection stage's current is split between the main and auxiliary source, with independent fractions K_1 and K_2 up to a maximum value K_{max}. The main current sources are still triggered internally by the respective previous stage of the oscillator.

To determine the phase alignment, assume the first stage's main current source turns on at $t = 0$. The main current source in the second stage turns on at $t = t_1$, triggered by the first stage. The injector current sources in both stages will be turned on at $t = t_d$ by the external trigger. The first stage can still be modeled by (6). The current in the second stage can be modeled with the following equations.

$$I_2(t) = \begin{cases} 0 & \text{if } t < t_1, t < t_d, \\ K_2 I_d & \text{if } t_d < t < t_1, \\ (1 - K_2) I_d & \text{if } t_1 < t < t_d, \\ I_d & \text{if } t \geq t_1, t \geq t_d \end{cases} \tag{7}$$

It can be assumed that the second stage must finish charging after the first stage finishes, so all the cases where $t_2 < t_1$ are extraneous. It can also be assumed that cases where $t_2 < t_d$ are extraneous. Otherwise, the injectors do not affect either stage triggering. Thus, all but the last case in (7) may be ignored, yielding the equation for t_2:

$$V_c(t_2) = V_{th} = \frac{I_d}{C} (t_2 - (1 - K_2) t_1 - K_2 t_d). \tag{8}$$

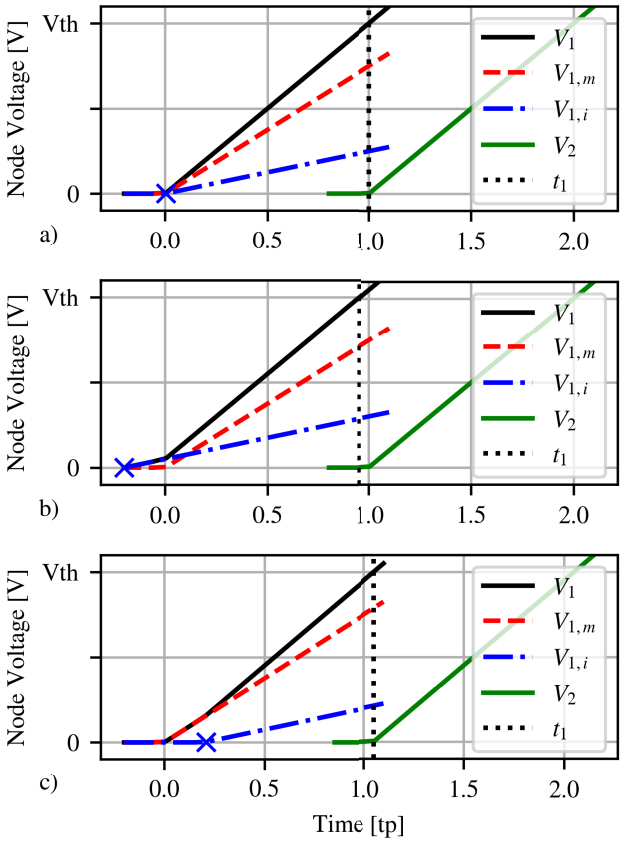


Fig. 6. Timing diagrams for the single-phase injection model (Fig 5 (b)). V_1 denotes the voltage in the first stage, and V_2 denotes the voltage in the second. The second stage charging is triggered when V_1 exceeds V_{th} at time t_1. $V_{1,m}$ denotes the contribution from the main current source in the first stage, and $V_{1,i}$ denotes the contribution from the injection source. The blue X denotes when the injection source turns on. In (a), the injection contribution aligns exactly with the main source, resulting in $t_1 = t_p$. (b) and (c) show the injection contribution starting early and late respectively. These cases both shift the timing of the next stage, changing the overall loop period.

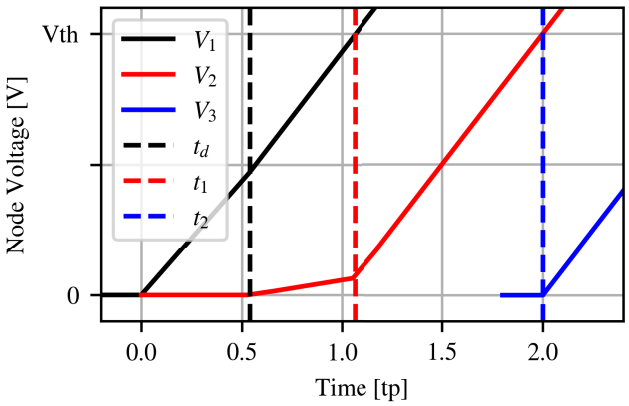


Fig. 7. Timing diagrams for the multi-phase injection model (Fig 5 (c)). V_1, V_2, and V_3 denote the voltages in the first, second, and third stages. In this example, equal injection is applied to both stages, i.e. $K_1 = K_2$. Although injection causes t_1 to arrive after $t = 1 \times t_p$, the combination of the effects still causes the second stage to arrive at $t_2 = 2 \times t_p$.

Recall that the free-running oscillator period is assumed to be the same as the injection period. To maintain this, the two adjacent injection stages should still contribute the same delay as any two non-injection stages. Therefore, $t_2 = 2t_p$. Combining this postulate with (6) and (8) yields the equation for t_d:

$$t_d = \begin{cases} \dfrac{K_2 - K_1}{K_2 - K_1 K_2} t_p & \text{if } t_1 < t_d \\ \dfrac{K_2}{K_1 + K_2 - K_1 K_2} t_p & \text{if } t_1 \geq t_d \end{cases} \tag{9}$$

Assuming $K_1 \geq 0$ and $K_2 \geq 0$, (9) contradicts (6) for $t_1 < t_d$, making the case extraneous. This results in the final equations.

$$t_d = \frac{K_2}{K_1 + K_2 - K_1 K_2} t_p \tag{10}$$

$$t_1 = \frac{K_1 + K_2}{K_1 + K_2 - K_1 K_2} t_p \tag{11}$$

Note that if $K_2 = 0$, then $t_d = 0$ and $t_1 = t_p$, meaning that injection aligns exactly with the start of the first stage and the first stage will contribute one stage delay. This matches the case with a single injection stage. Conversely, if $K_1 = 0$, then $t_d = t_p$, meaning that injection aligns exactly with the start of the second stage.

Fig 8 demonstrates an example using this model. Fig 8(a) presents example values for K_1 and K_2 over some discrete codes, using the simple scheme where K_1 linearly decreases while K_2 linearly increases. This causes the shift from aligning to the first stage to aligning to the second stage. Fig 8(b) shows the resulting injection phase from these K values. The reference shows an exactly linear case where each code step is equal. The results from the model show a clear deviation from the reference line. Fig 8(c) shows the impact on the first stage's delay. Ideally, the stage always contributes the unaffected propagation delay t_p. However, this MPI method does induce a static error.

C. Extension to the Full Oscillator

To derive the static phase rotation nonlinearity from this method, the phase must be defined relative to a fixed reference, in this case, the injection signal. The following equations define the phase integral non-linearity (INL).

$$\phi_k[n] = t_k[n] - t_d[n] \tag{12}$$

$$\phi_{ideal}[n] = \frac{n}{N} \tag{13}$$

$$INL_k[n] = \phi_k[n] - \phi_{ideal}[n] \tag{14}$$

where k denotes the observed oscillator stage, the code n defines K_1 and K_2, N is the total number of codes, and t_{per} is the oscillator period.

t_d and t_1 were derived in the previous section, and $t_2 = 2t_p$ was defined. Because the remaining stages have no injection, $t_3 = 3t_p$, $t_4 = 4t_p$, etc. These and the previously derived equations hold while injection is applied to the first and second stages. Once injection shifts fully off the first stage and all onto the second stage, the new adjacent injection pair will be the second and third stages (Fig. 9) and so on. Under these new conditions, t_1 will appear unperturbed, i.e., $t_1 = t_p$. 10 and 11

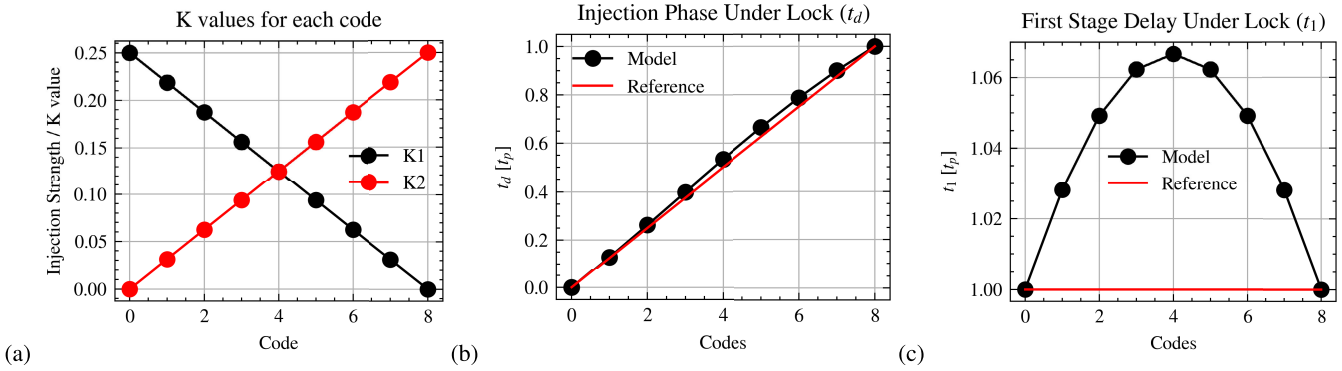


Fig. 8. Example results from the model. (a) Example K values for K_1 and K_2 (b) Resulting injection phase under lock t_d (c) Resulting first stage delay t_1.

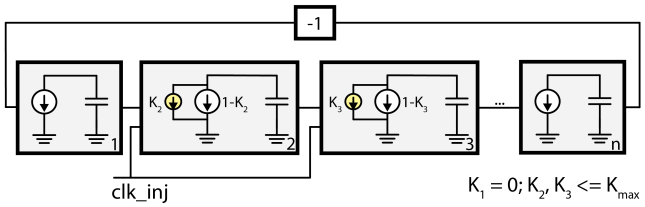


Fig. 9. After K_1 goes to 0 in Fig 5(c), injection shifts to the next pair.

can thus be generalized as the injection pair changes.

$$t_d = \frac{K_i}{K_i + K_{i+1} - K_i K_{i+1}} t_p + (i-1)t_p \tag{15}$$

$$t_1 = \begin{cases} \dfrac{K_1 + K_2}{K_1 + K_2 - K_1 K_2} t_p & \text{if } K_1 > 0 \ \& \ K_2 > 0 \\ t_p & \text{else} \end{cases} \tag{16}$$

where i and $i+1$ are the current injection pair. These can be plugged into (14) to compute the induced phase error and INL. Fig. 10 shows an example of this error for an 8-phase oscillator with single-phase injection using the values from Fig. 8. The units are given in propagation delays (t_p) and in degrees (°), with the latter being agnostic to both the oscillator period and the number of oscillator phases. The error in the first eight codes notably has a different sign than the latter periodic codes. During the first eight codes, MPI is applied to the observed stage, so the effects of shifting t_1 (as in Fig. 8(c) appear. In the latter codes, we only see the effect from injection phase lock (as in Fig. 8(b).

The proposed circuit is an 8-phase oscillator with differential injection and 64 rotation codes. Fig. 11 shows the modeled static phase INL for this case. Additionally, the INL is shown for multiple values of K_{max}, the maximum injection strength for K_1 and K_2. The values for K_1 and K_2 are similar to Fig. 8(a) where one linearly increases while the other decreases. This shows that the worst-case phase error is improved by a smaller K_{max}. Therefore, for phase resolution not to be limited by the phase error induced by MPI, K_{max} needs to be reduced. However, this will also shrink the injection oscillator's phase noise filtering bandwidth, increasing jitter [18]. Thus, there is a trade-off between jitter and resolution. This work uses a larger K_{max} to meet the jitter specification while maximizing power efficiency and achieving

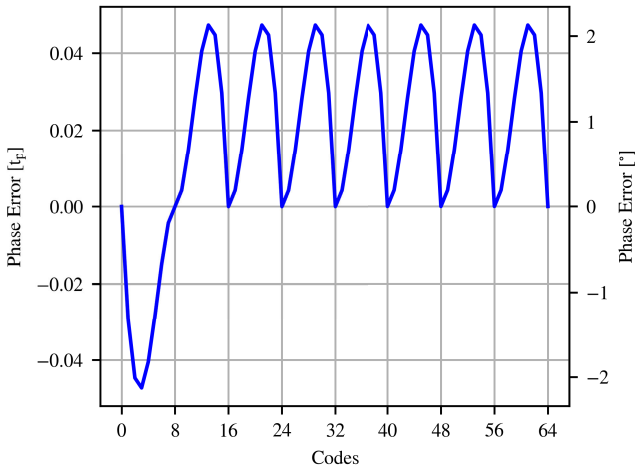


Fig. 10. MPI-induced phase error for an 8-phase oscillator with a single injection phase. The results in degrees (right axis) are agnostic to both the oscillator period and the number of oscillator phases.

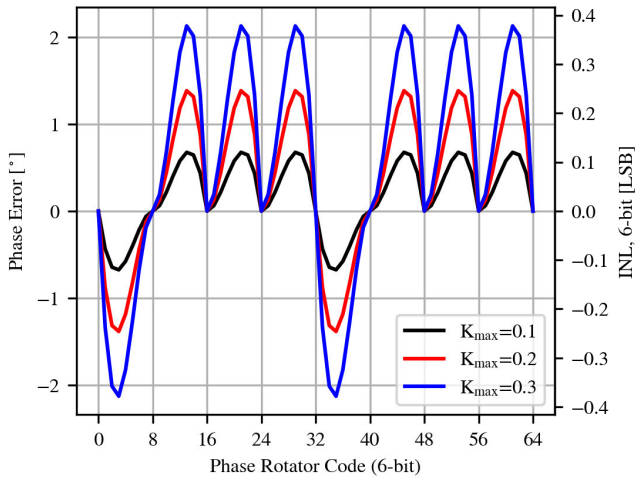


Fig. 11. MPI-induced phase error for an 8-phase oscillator with differential injection. The error is also reported as INL, with a 6-bit LSB based the proposed circuit. Results for varying values of K_{max} show the impact of K_{max} on static phase error.

better than 6 bits of resolution. The exact value for K_{max} was chosen to yield simple ratios for device and current sizing, easing layout effort.

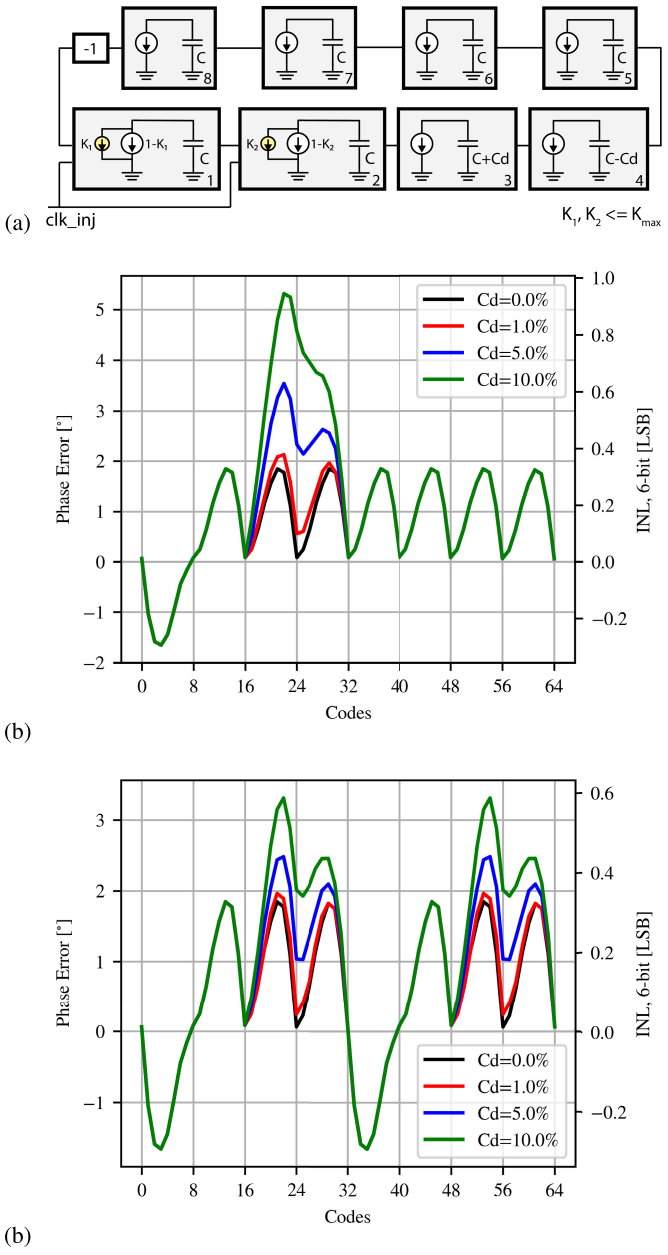


Fig. 12. (a) Extended model of an 8-phase MPI oscillator with stage mismatch introduced through capacitor variation. (b) Model results for single-ended injection (c) Model results for differential injection.

D. Modeling Mismatch in Stage Delay

The discussion thus far has assumed no delay mismatch between the oscillator stages. Delay mismatch can be introduced by affecting the circuit parameters. Fig. 12(a) shows an example of this by modifying the capacitor values in an 8-phase oscillator. The prior derivation can be reused with one major modification: a pair of stages can no longer be assumed to contribute $2t_p$. This can be seen, for example, with stages 2 and 3 in Fig. 12 (a). The more generalized postulate is that the oscillator loop delay should be constant. In Fig. 12(a), the loop delay is $8t_p$, and the delay mismatch from stage 3 is compensated for by stage 4.

The computed phase error for the single-ended case is shown in Fig. 12(b) for varying values of mismatch. The additional error is isolated to the stages with the mismatch. Larger values of mismatch significantly impact the linearity, even exceeding the error introduced by the proposed MPI technique. Fig. 12(c) further shows the error for differential injection with mismatch only in stages 3 and 4. The differential natures causes the error from the first half of the codes to repeat in the second half. The peak-to-peak error is less in the differential case versus the single-ended. However, if the same mismatch is also introduced into stages 7 and 8, the peak-to-peak error is the same.

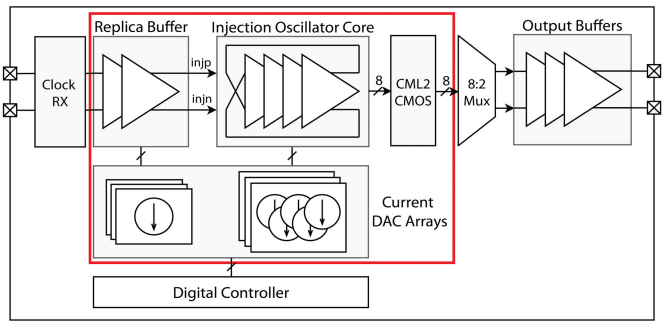


Fig. 13. Implemented phase rotator system architecture. The key contributions are boxed in red.

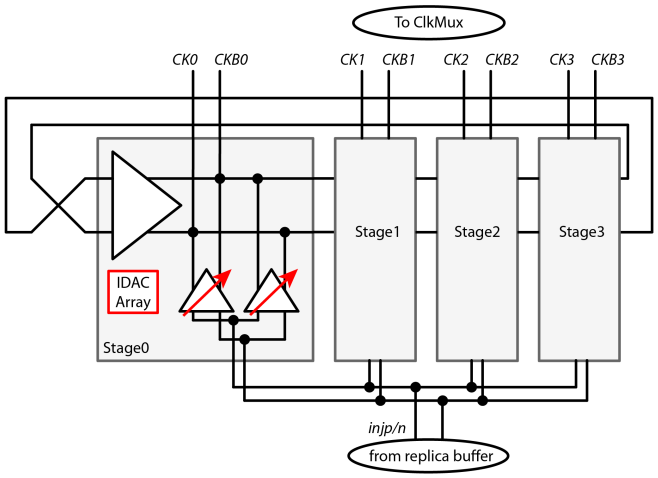


Fig. 14. Injection oscillator core. The differential injection clock is available to each stage. Injection strength into each stage is controlled by the current DAC arrays. The current DAC code is set by the digital phase controller.

III. Circuit Implementation

The complete implemented system is presented in Fig. 13. The core system consists of the injection oscillator core, the current DAC arrays, the replica input buffers, and the output buffers. This section discusses these key circuit components.

A. Injection Oscillator Core

Fig. 14 shows the injection oscillator core. The main oscillator is a 4-stage differential (8-phase) current-mode-logic (CML) based ring oscillator. The current-mode design matches the proposed model well, allowing the model to be used as the reference for the design. It also more easily achieves the high target frequency compared to a similar voltage-mode design.

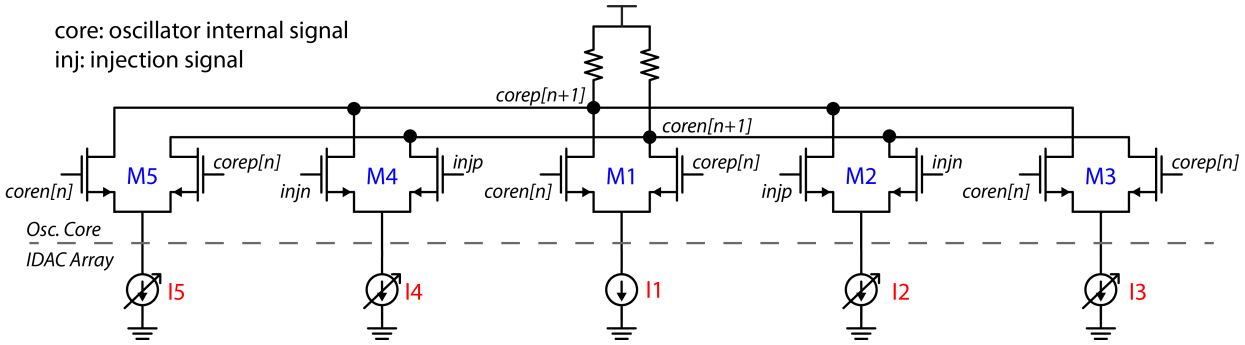


Fig. 15. Oscillator Stage. M2 and M4 are driven by the injection signal, while M1, M3, and M5 are driven by the oscillator internal signal.

While using external clock muxing or flipping has been proposed to provide the differential phase selection with hardware reuse [19], this introduces an additional tuning knob and may introduce spurs during dynamic operation. Therefore, the differential selection is kept internal to the oscillator. The internal selection allows for a single, smooth tuning knob for phase selection at the cost of higher oscillator core power.

B. Oscillator Stage

The oscillator delay stage is shown in Fig. 15. Each delay stage contains 5 NMOS differential pairs. The primary pair (M1) is driven by the oscillator's internal signal and is always on. One pair (M2) is driven by the differential injection signal, and another (M4) is driven by the injection signal with a flipped polarity for injection polarity selection. These injection pairs are never on at the same time. Additionally, each of these injection pairs is complemented with another pair connected to the oscillator's internal signal (M3 & M5). When a given injector (e.g. M2) is turned off, the complement (M3) is turned on, allowing each stage to maintain the same current regardless of rotator code.

The resistive loads are implemented by using passive precision resistors for ease of design, due to their constant impedance over the operating range. While a Maneatis cell [20] or similar loads can provide highly linear impedance over the operating range, they do not significantly improve on the area in this design and require an additional bias generation circuit, including a biasing DAC.

Given the small set of available parameters - transistor width, resistor size, and bias current - and impact of layout parasitics, the design procedure for the oscillator core is fundamentally straightforward: sweep the resistor and transistor sizes to meet frequency for a given bias, then find the bias the yields the lowest jitter. Intuitively, for power efficiency, this results in maximizing the transistor width to operate at a low V*. In addition to better jitter filtering, the larger K_{max} provides a smaller ratio between injection device (M2, M4) and oscillator device (M1). This smaller ratio helps reduce the dynamic range of the device sizes, making the oscillator core smaller overall and more efficient.

C. Current DAC Arrays

Each oscillator stage requires a current DAC array to generate the five currents (I1-I5 in Fig. 15). Fig. 16 shows the structure of such an array. The unit cells in the DACs feature stacked transistors to achieve lower current per unit cell, allowing the DAC to achieve high current resolution. The enable gates for I1 are tied high to keep them always on. As unit cells for I2 are turned off, I3 unit cells are turned on to maintain a constant total current. I4 and I5 are produced with arrays identical to I2 and I3 respectively.

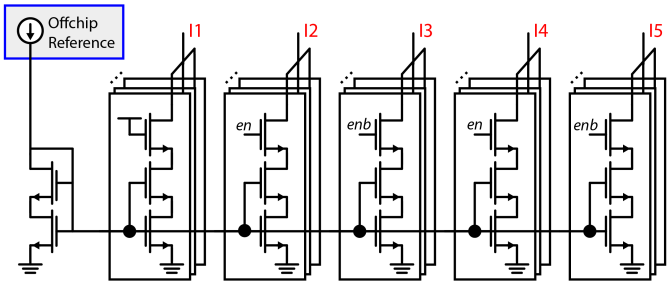


Fig. 16. Current DAC bank for a single stage. I1 is always on. As unit cells for I2 are turned off, I3 unit cells are turned on to maintain constant total current. I2 and I4 are never on at the same time.

The injection strength is set by the current ratio between the NMOS pairs driven by injection and the pairs driven by the oscillator's signal up to a maximum injection ratio K_{max}. In this design, each current DAC (I2, I3, I4, and I5) has 8 unit cells for injection control. The always-on DAC (I1) has 16 unit cells. To perform MPI, the injection into one stage linearly decreases while the injection in the next stage linearly increases, similar to the method shown in Fig. 8(a).

D. Inputs and Output Buffers

Fig. 17 shows the input buffer system. A 2-stage replica buffer is included before the oscillator core to condition the injection signal to look like the oscillator's CML signal. Each buffer stage is sized the same as the oscillator core and consumes current from a similar always-on current DAC. In simulation, compared to using only an AC-coupling input stage, adding two stages of replica buffer improves the INL by 3 or more degrees (equivalent to 0.5 LSB or more for this 6b design), depending on the amplitude of the reference signal. The trade-off is additional power and a slight increase in random jitter due to the noise of the additional buffer stages. At the input to the replica buffer is a high-pass filter for level

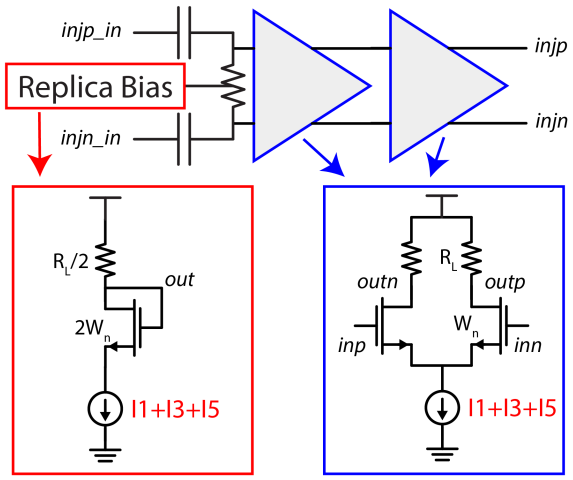


Fig. 17. Replica buffer and bias circuit.

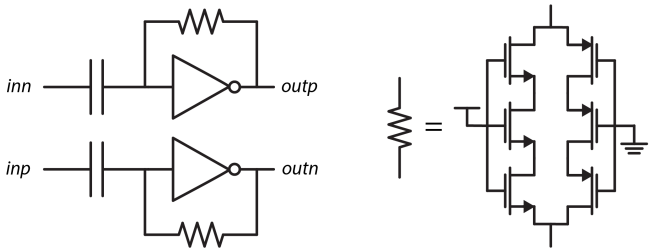


Fig. 18. CML2CMOS buffers and resistor implementation.

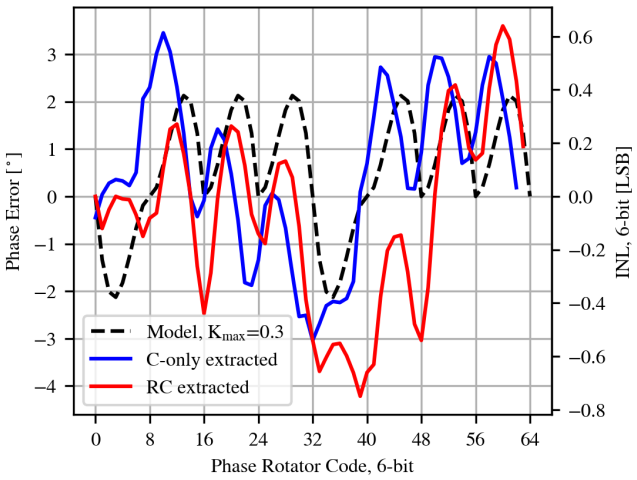


Fig. 19. INL comparison between model, C-only extracted, and RC-extracted circuit simulations.

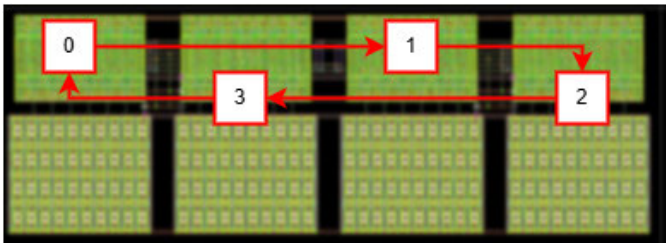


Fig. 20. Layout of the oscillator core. The indices of each differential stage and the routing order are labeled. The routing differences, e.g., between the route from stage 0 to 1 and the route from stage 1 to 2, contributed to discrepancies between the modeled INL and realized circuit's INL.

shifting the injection signal. The high-pass filter is biased by using a replica bias.

Following the oscillator core are CML2CMOS buffers for bringing the low CML swing back to full CMOS swing, mirroring a real environment where the generated clocks will be distributed or further conditioned, e.g., with duty cycle correction or fine delay correction. The CML2CMOS buffers are AC-coupled self-biased inverter amplifiers (Fig. 18) and are preferred for their low power. The capacitors are implemented by using passive metal-on-metal (MOM) capacitors, and the resistors are implemented as stacked transistors for compactness.

Simulation results for the complete system, including the replica input buffers and CML2CMOS output buffers, are shown in Fig. 19. The expected INL from the model is also shown. The simulation shows a good correlation with the model. Deviation from the model is attributed to layout asymmetries, as highlighted in Fig. 20.

E. Auxiliary Circuits

Due to packaging limitations, only two phases are observable from the chip. An 8:2 clock mux is included after the CML2CMOS buffers to support this. This mux consists of 2 stages of passgate-based muxes with inverter buffers. Finally, the selected clocks are buffered and fed to a CML clock driver to go off-chip. For testing, a digital controller is included to enable or disable injection, statically set the phase code, or enable a continuously incrementing phase code using an external clock.

IV. Circuit Generation

Analog and mixed-signal (AMS) circuit designers face challenges with design iterations in advanced process nodes due to higher sensitivity to parasitics and increasingly complex design rules. Circuit generators address this issue by codifying a designer's intent for how a circuit is sized and laid out, reducing iteration time between schematic and layout. The generator consumes technology-specific parameters and designer-given specifications to produce a DRC- and LVS-clean layout and schematic. The Berkeley Analog Generator (BAG) performs this using an open-source Python framework for scripting circuit generation, simulation management, and design optimization [21], [22], [23].

The AMS components of this proposed clock generator are all designed in BAG3++, the latest version of BAG. The BAG-generated blocks and associated layout generators are shown in Fig. 21. Each layout generator builds upon an appropriate *XBase* layout template: *MOSBase* for dense transistor layouts, *ResArrayBase* for tiling resistors, and *TemplateBase* for assembling subblock [22], [24]. The generator code then specifies how the base transistor or resistor grid is drawn, where specific devices are placed (e.g., for matching or interleaving), and how the circuit is routed. Finally, a schematic is produced by using

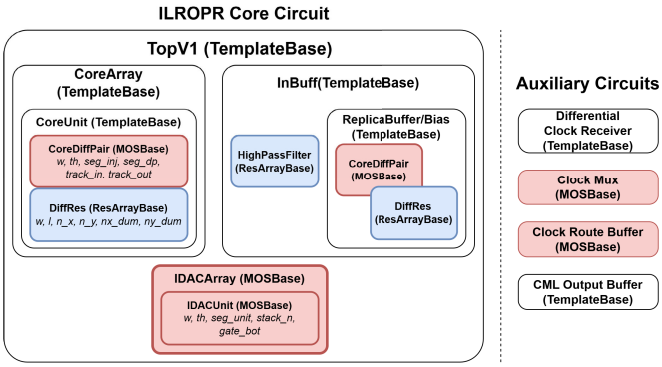


Fig. 21. High-level layout generator structure. Dense transistor blocks (MOSBase) are in red, resistor arrays (ResArrayBase) are in blue, and general layouts (TemplateBase) are in white. Example parameters are shown, including both sizing or schematic parameters (*w_n*, *seg_inj*, etc.) and layout-specific parameters (*track_in*, *track_out*).

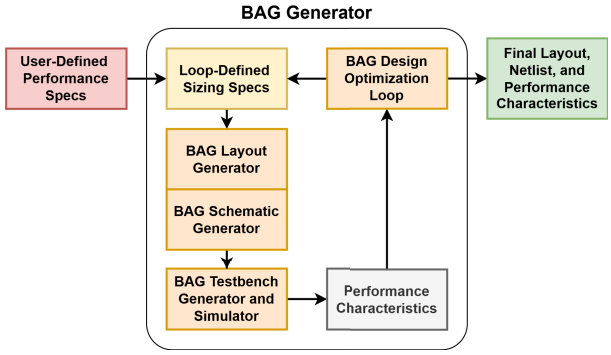


Fig. 22. Typical BAG design flow.

a schematic circuit generator and schematic parameters produced by the layout generator. The technology-agnostic layout generators have been released under open-source licenses [25].

A typical generator design flow involves decomposing input specifications to block-level parameters then co-iterating on subblocks and integration to meet the specifications (Fig. 22). This has been exemplified in prior work [23], [26], [27]. As the key circuit here - the oscillator core - has relatively few parameters (resistor and transistor size), a specification decomposition is not required. Instead, a relatively simple design loop was used for sizing the core: given a target bias current, sweep the resistor parameters and find the transistor size to meet the frequency. Then, given the found design space, return the design with the lowest jitter. The replica buffer and bias circuits are then sized to match the core circuit. Note that the injection strength K_{max} is an architectural parameter and so it and the associated INL performance are not included in the design loop. Using this generator-based flow provides tape-out-ready layouts from simple sizing changes. This allows for real parasitics to be accounted for at design time, which is critical to high-frequency design.

Once realized, the generated oscillator core layout is combined in BAG with the current DACs and buffers along with additional test circuits, namely the clock receiver and output clock mux. It is then manually integrated with other top-level components, including the digital controller (implemented

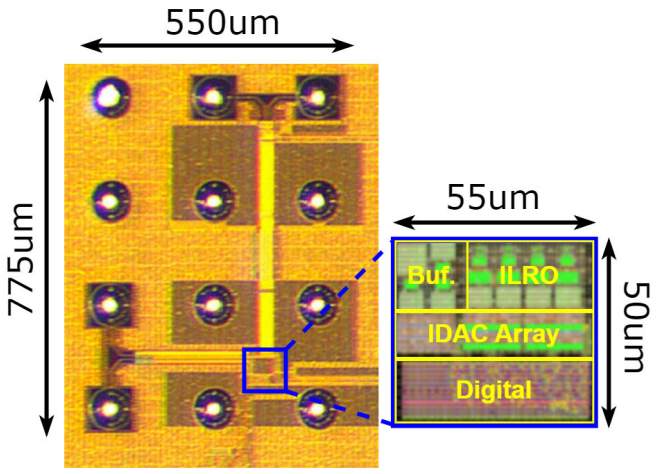


Fig. 23. Die micrograph with details of the implemented layout.

using automatic place-and-route), top-level routing and power grid, ESD, and IO buffers.

V. Measurements

The proposed design was fabricated in the Intel 16 process [28], targeting a clock frequency of 17 GHz. The analog core area is 55 μm $\times$ 30 μm, and the digital core is 55 μm $\times$ 20 μm. For comparison, to achieve the same functionality with CMOS phase interpolators requires an analog core area of 53 μm $\times$ 60 μm. This proposed architecture presents a 50% analog area reduction. The measured analog power consumption, including the oscillator core, rotator, and input and output buffers, is 33 mW from a 1.2V supply.

The measurement setup is shown in Fig. 24. The chip is flip-chip packaged with a custom organic interposer and then soldered to a Rogers 4350B test board. A Keysight E8257D clock signal source is used as a clock reference. This clock is first split through a power divider. One output of the power divider goes to a high-frequency balun, generating the differential reference clock, and to the chip through DC blocks. The chip output comes out through DC blocks to either a Keysight N9010A spectrum analyzer for spectrum and phase noise measurements, or to a Keysight N1000A DCA-X sampling scope for clock phase and linearity measurements. The second output of the power divider also goes to the sampling scope as a reference. DC power to the chip is controlled using I^2C to LDOs on a power-conditioning board. Current references for the oscillator's current DACs come from benchtop SMUs.

Fig. 25 shows the measured output frequency versus the reference injection frequency under injection lock. The oscillator exhibits a wide locking range due to the selected high K_{max}. While 17 GHz is not the exact center frequency of the injection range, it still yields the best linearity results in the subsequent measurements.

Fig. 26(a) shows the MPCG phase-to-phase (0°-45°) error. At a fixed rotator code, the 8-phase error was measured to be under 17 degrees (°) peak-to-peak across all phases. The source of this large error is only partly attributable to random process variation, since Monte Carlo simulations reported a

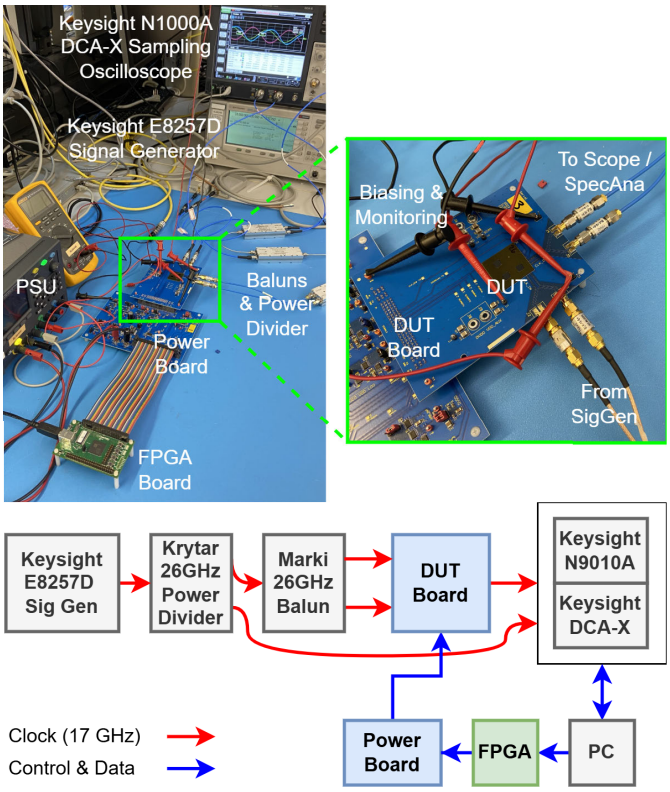


Fig. 24. Measurement setup.

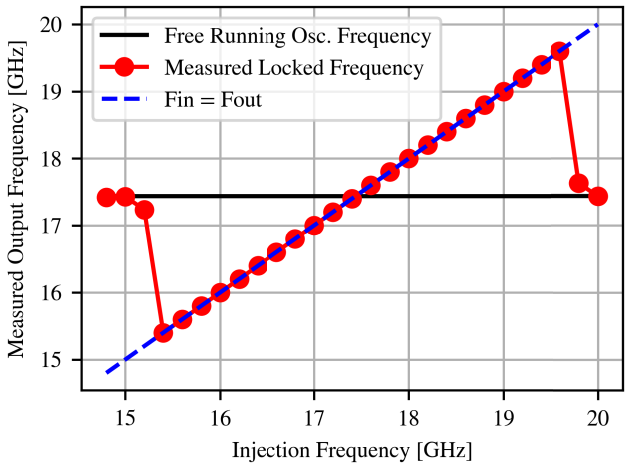


Fig. 25. Measured output frequency versus injection frequency.

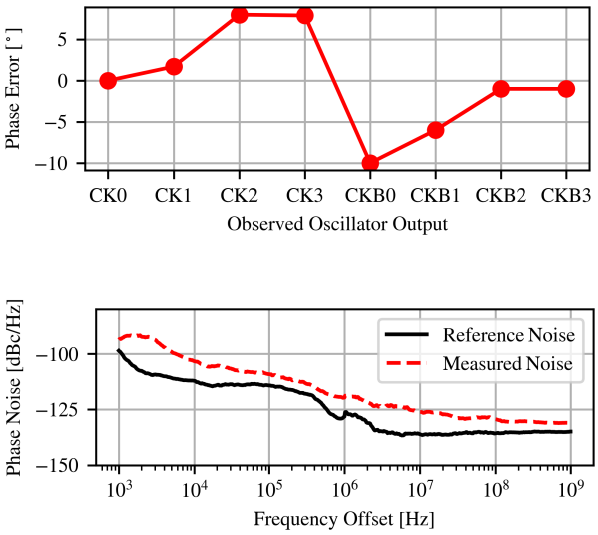


Fig. 26. (a) Measured MPCG error and (b) phase noise spectrum.

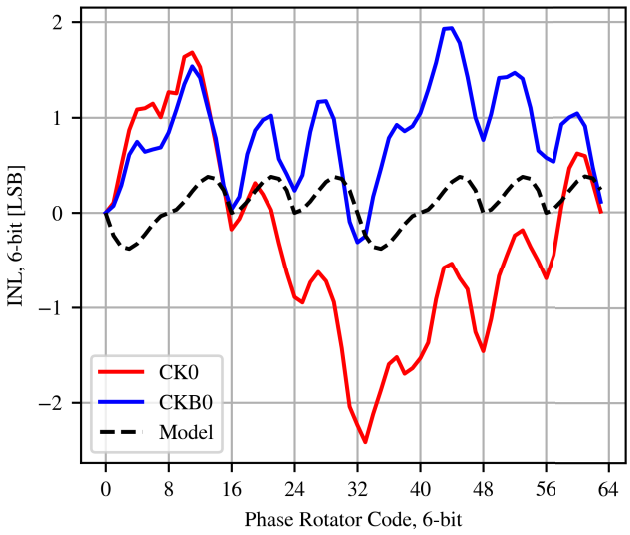


Fig. 27. Measured best (CKB0) and worst (CK0) INL curves with the mismatch-free modeled INL.

phase error standard deviation of less than 1°. Fig. 26(b) shows the measured phase noise spectrum. The phase noise at 1 MHz offset frequency is -119.6 dBc/Hz. The measured integrated RMS jitter from 10KHz to 1GHz is 98 fs. The jitter varies by less than 3 fs over the rotator codes. This is expected since the current into each stage is constant across all codes.

INL was measured for each of the 8 output phases. The best and worst curves are shown in Fig. 27. Generally, the INL curve contains the expected systematic behavior from modeling. However, the size of the error peaks exceeds expectations by approximately a factor of 2. The errors are primarily attributed the MPCG phase-to-phase errors from Fig. 26(a), following the mismatch model results in Fig. 28. The mismatch model results also suggest that the differential injection may not be working correctly. Additional errors may be attributed uncaptured top-level layout parasitics. Fig. 29 summarizes the DNLpp and INLpp for all outputs. For this 6b design, the worst static DNLpp and INLpp are 1.26 LSB and 4.05 LSB respectively. The considerable variation in linearity between outputs suggests some other systematic physical issue.

Dynamic performance was measured by observing the output spectrum while incrementing the rotator phase code. The code incrementation was controlled by the digital controller and clocked up to 48 MHz. The largest spur varies between −27.4 and −28.8 dBc at +/− 0.75 MHz offset, depending on the selected output phase. Fig. 30(a) shows this measured spectrum and highlights the largest spurs. These large spurs correlate with the discrete Fourier transform (DFT) of the INL

TABLE I

PERFORMANCE SUMMARY AND COMPARISON

		PR-based			PI-based	
Metric	**This Work**	O'Mahony [10]	Hossain [12]	Huang [29]	Wang [30]	Chen [31]
Architecture	MPI-ILO-PR	MPI-ILO-PR	MPI-ILO-PR	MPI-ILO-PR	MPI-ILO + PI	QLL + PI
Process	Intel 16 [28]	45nm	65nm	16nm	65nm	7nm
Freq [GHz]	17	4.05	3.7	7	7	16
Number of Phases	8	8	2	8	2	2
Jitter [fs RMS]	98	1700	1400	254	84	80
Number of Bits	6	7	7	8	7	7
Resolution [ps]	0.92	1.93	2.11	0.56	1.12	0.49
INLpp [LSB]	4.05	2.20[b]	-[a]	1.14	1.72	1.74
DNLpp [LSB]	1.26	0.70[b]	0.90[b]	1.31	1.13	0.87
Supply [V]	1.2	1	1	1.2	1.2	1.2
Power [mW]	33.6	14.3	4.4	11.4	22.95[c]	22.4
PLL FOM	-244.9	-223.8	-230.6	-241.3	-247.9	-248.4
MPCG FOM	-266.2	-238.9	-239.3	-258.8	-259.4	-263.5

[a]Not reported. [b]Estimated from charts. [c]Includes MPCG and PI.

$\text{FOM}_{\text{PLL}} = 10\log_{10}\left((\text{Jitter}_{\text{RMS}})^2\left(\frac{\text{Power}}{1\text{mW}}\right)\right)$

$\text{FOM}_{\text{MPCG}} = 10\log_{10}\left((\text{Jitter}_{\text{RMS}})^2\left(\frac{\text{Power}}{1\text{mW}}\frac{1\text{GHz}}{\text{Frequency}}\frac{1}{\text{Num. Phases}}\right)\right)$

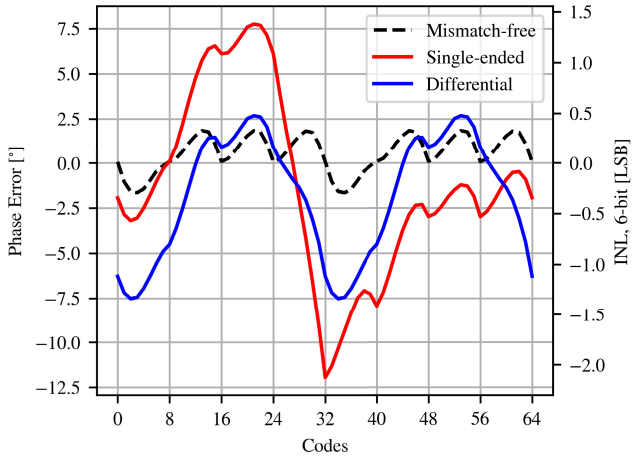
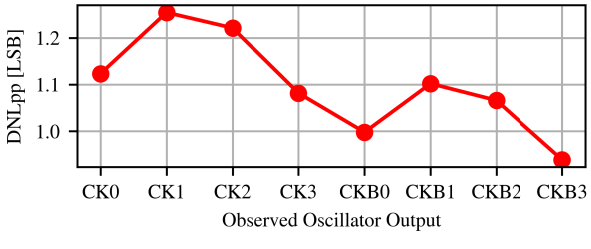
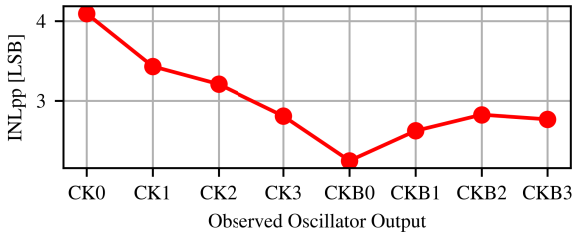


Fig. 28. Modeled INL including the MPCG phase error from Fig 26(a), for both single-ended and differential injection. The single-ended case bears correlation with the performance in Fig 27.

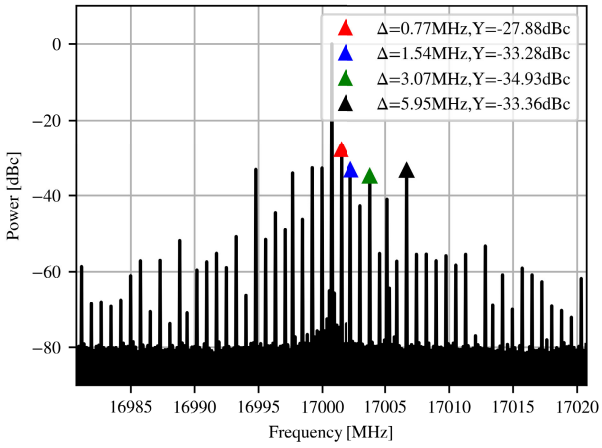


(a)

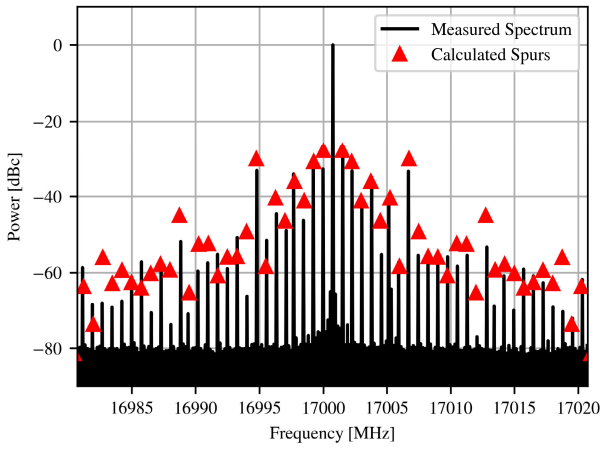


(b)

Fig. 29. (a) Measured DNLpp and (b) INLpp for all clock outputs.



(a)



(b)

Fig. 30. Spectrum for one of the output phases, measured while incrementing the rotator code at 48 MHz. (a) highlights the major spurs at increments of roughly 48 MHz / 64 = 0.75 MHz. (b) shows the expected spurs calculated from the DFT of the measured static INL. The results show good correlation.

characteristic, as shown in Fig. 30(b). The integrated fractional spur is −24.0 dBc.

Table I compares the performance of the proposed circuit with other multi-phase clock generators. Ref. [10], [12], [29] demonstrate similar phase rotator-based techniques, while [30], [31] demonstrate PI-based techniques. Compared to prior phase rotator-based work, this work demonstrates the best jitter (PLL) FOM and multi-phase generation efficiency (MPCG FOM) while demonstrating high resolution rotation. Compared to the PI-based work, this work presents comparable jitter performance and phase resolution. By performing simultaneous rotation, it achieves better MPCG multi-phase generation efficiency.

VI. Conclusion

This paper describes the design techniques for a high frequency (≥16 GHz) multi-phase clock generator with efficient built-in phase rotation. Multi-phase injection is used to perform high-resolution reference-side rotation, efficiently rotating all generated phases. A detailed model demonstrates the impact of injection strength on the static nonlinearity. The circuits in this work were generated using BAG3++, allowing for layout-aware design iteration and optimization. This phase rotator technique achieves better MCPG efficiency than PI-based techniques at comparable frequencies. By using a static injection method, the proposed technique also significantly improves jitter over prior phase rotators. Future work on this rotator method may investigate techniques for frequency locking using a quadrature phase detector (QPD) and implementation using a fully CMOS architecture.

Acknowledgment

The authors would like to thank the BWRC member companies, students, staff, and faculty for their support. They acknowledge the Intel University Shuttle Program for donation of test chip fabrication.

References

[1] *PCI-SIG Announces PCI Express 7.0 Specification to Reach 128 GT/s*. Accessed: Nov. 2024. [Online]. Available: https://www.businesswire.com/news/home/20220621005137/en

[2] *Adopted IEEE P802.3dj Objectives*. Accessed: Nov. 2024. [Online]. Available: https://www.ieee802.org/3/dj/projdoc/objectivesP802d3dj240314.pd

[3] H. Lin et al., "A 4×112 Gb/s ADC-DSP based multistandard receiver in 7nm FinFET," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2020, pp. 1–2, doi: 10.1109/VLSICircuits18222.2020.9162802.

[4] T. Ali et al., "6.2 A 460 mW 112Gb/s DSP-based transceiver with 38 dB loss compensation for next-generation data centers in 7 nm FinFET technology," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 118–120, doi: 10.1109/ISSCC19947.2020.9062925.

[5] H. Park et al., "A 4.63pJ/b 112Gb/s DSP-based PAM-4 transceiver for a large-scale switch in 5 nm FinFET," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 5–7, doi: 10.1109/ISSCC42615.2023.10067613.

[6] Z. Guo et al., "A 112.5Gb/s ADC-DSP-based PAM-4 long-reach transceiver with >50dB channel loss in 5 nm FinFET," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 116–118, doi: 10.1109/ISSCC42614.2022.9731650.

[7] D. Pfaff et al., "7.3 A 224Gb/s 3pJ/b 40 dB insertion loss transceiver in 3 nm FinFET CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 67, Feb. 2024, pp. 128–130, doi: 10.1109/ISSCC49657.2024.10454537.

[8] Y. Segal et al., "A 1.41pJ/b 224Gb/s PAM-4 SerDes receiver with 31dB loss compensation," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 114–116, doi: 10.1109/ISSCC42614.2022.9731794.

[9] J. Q. Wang et al., "7.1 A 2.69pJ/b 212Gb/s DSP-based PAM-4 transceiver for optical direct-detect application in 5nm FinFET," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 67, Feb. 2024, pp. 123–125, doi: 10.1109/ISSCC49657.2024.10454275.

[10] F. O'Mahony, B. Casper, M. Mansuri, and M. Hossain, "A programmable phase rotator based on time-modulated injection-locking," in *Proc. Symp. VLSI Circuits*, Jun. 2010, pp. 45–46, doi: 10.1109/VLSIC.2010.5560263.

[11] Y. F. Zhang, J. Liang, and T. C. Carusone, "Design considerations for time-modulated injection-locked phase interpolators and rotators," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 1719–1723, doi: 10.1109/ISCAS48785.2022.9937957.

[12] M. Hossain and A. Chan Carusone, "7.4 Gb/s 6.8 mW source synchronous receiver in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1337–1348, Jun. 2011, doi: 10.1109/JSSC.2011.2131730.

[13] B. Zhou and B. Nikolić, "A multi-phase injection-locked 8-phase 17 GHz clock generator with 6b phase rotation for multi-phase wireline receivers," in *Proc. IEEE Eur. Solid-State Electron. Res. Conf. (ESSERC)*, Sep. 2024, pp. 629–632, doi: 10.1109/ESSERC62670.2024.10719579.

[14] Y. Zhang, Z. Wang, and P. R. Kinget, "Analysis of injection-locked ring oscillators for quadrature clock generation in wireline or optical transceivers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 8, pp. 3074–3082, Aug. 2022, doi: 10.1109/TCSI.2022.3175111.

[15] Z. Wang, "AM-to-PM conversion in injection-locked, inverter-based ring oscillators," in *Efficient and High-Performance Clocking Circuits for High-Speed Data Links*. New York, NY, USA: Columbia Univ., 2022, ch. 5.2.1.

[16] J.-C. Chien and L.-H. Lu, "Analysis and design of wideband injection-locked ring oscillators with multiple-input injection," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1906–1915, Sep. 2007, doi: 10.1109/JSSC.2007.903058.

[17] G. R. Gangasani and P. R. Kinget, "Time-domain model for injection locking in nonharmonic oscillators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1648–1658, Jul. 2008, doi: 10.1109/TCSI.2008.916605.

[18] B. Razavi, "A study of injection locking and pulling in oscillators," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1415–1424, Sep. 2004, doi: 10.1109/JSSC.2004.831608.

[19] Z. Wang and P. R. Kinget, "A 65 nm CMOS, 3.5-to-11 GHz, less-than-1.45LSB-INLpp, 7b twin phase interpolator with a wideband, low-noise delta quadrature delay-locked loop for high-speed data links," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 65, Feb. 2022, pp. 292–294, doi: 10.1109/ISSCC42614.2022.9731649.

[20] J. G. Maneatis and M. A. Horowitz, "Precise delay generation using coupled oscillators," *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1273–1282, Dec. 1993, doi: 10.1109/4.262000.

[21] J. Crossley et al., "BAG: A designer-oriented integrated framework for the development of AMS circuit generators," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 74–81, doi: 10.1109/ICCAD.2013.6691100.

[22] E. Chang et al., "BAG2: A process-portable framework for generator-based AMS circuit design," in *Proc. Cust. Integr. Circuits Conf. (CICC)*, Apr. 2018, pp. 1–8.

[23] F. Guo et al., "Bag3++: An extensible generator framework for automated layout-aware ams design," *IEEE Open J. Circuits Syst.*, to be published.

[24] (2024). *Xbase*. [Online]. Available: https://github.com/ucb-art/xbase

[25] (2024). *BAG ILROPR*. [Online]. Available: https://github.com/ucb-art/

[26] Z. Liu and B. Nikolić, "A generated 4 GS/s 124.6 mW 8× time-interleaved SAR-VCO ADC with 9.1 ENOB," in *Proc. IEEE Eur. Solid-State Electron. Res. Conf. (ESSERC)*, Sep. 2024, pp. 85–88, doi: 10.1109/ESSERC62670.2024.10719498.

[27] Z. Wang et al., "An output bandwidth optimized 200-Gb/s PAM-4 100-Gb/s NRZ transmitter with 5-tap FFE in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 21–31, Jan. 2022, doi: 10.1109/JSSC.2021.3109562.

[28] Q. Yu et al., "mmWave and sub-THz technology development in Intel 22nm FinFET (22FFL) process," in *IEDM Tech. Dig.*, Dec. 2020, pp. 17.4.1–17.4.4, doi: 10.1109/IEDM13553.2020.9372105.

[29] Y.-C. Huang and B.-J. Chen, "30.7 An 8b injection-locked phase rotator with dynamic multiphase injection for 28/56/112Gb/s serdes application," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 486–488, doi: 10.1109/ISSCC.2019.8662292.

[30] Z. Wang, Y. Zhang, Y. Onizuka, and P. R. Kinget, "11.4 A high-accuracy multi-phase injection-locked 8-phase 7GHz clock generator in 65 nm with 7b phase interpolators for high-speed data links," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 186–188, doi: 10.1109/ISSCC42613.2021.9365800.

[31] S. Chen et al., "A 4-to-16GHz inverter-based injection-locked quadrature clock generator with phase interpolators for multi-standard I/Os in 7 nm FinFET," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2018, pp. 390–392, doi: 10.1109/ISSCC.2018.8310348.

Bob Zhou (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2018, where he is currently pursuing the Ph.D. degree with Berkeley Wireless Research Center.

He has previously worked with Apple and Blue Cheetah Analog Design. His research interests include wireline link architecture and modeling, high-speed AMS design, and design automation and optimization.

Borivoje Nikolić (Fellow, IEEE) received the Dipl.-Ing. and M.Sc. degrees in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 1992 and 1994, respectively, and the Ph.D. degree from the University of California at Davis, Davis, CA, USA, in 1999.

In 1999, he joined the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, where he is currently a Professor and holds a National Semiconductor Distinguished Professorship in Engineering.

Dr. Nikolić received the NSF CAREER Award in 2003, the College of Engineering Best Doctoral Dissertation Prize, the Anil K. Jain Prize for the Best Doctoral Dissertation in Electrical and Computer Engineering at the University of California at Davis in 1999, and the City of Belgrade Award for the Best Diploma Thesis in 1992. For work with his students and colleagues, he has received the best paper awards at the IEEE International Solid-State Circuits Conference, the Symposium on VLSI Circuits, the IEEE International SOI Conference, European Solid-State Device Research Conference, European Solid-State Circuits Conference, the IEEE Custom Integrated Circuits Conference, the S3S Conference, the Design Automation Conference, and the ACM/IEEE International Symposium of Low-Power Electronics. He was a Technical Program Chair for the 2022 Symposium on VLSI Technology and Circuits and the General Chair for the 2024 IEEE Symposium on VLSI Technology and Circuits. From 2014 to 2015, he was a Distinguished Lecturer of the IEEE Solid-State Circuits Society.