2021 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-7281-9201-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/ISCAS51556.2021.9401383

A Power-efficient and Fast-locking Digital Quadrature Clock Generator with Ping-pong Phase Detection

Heon Hwa Cheong[1,2] and Suhwan Kim[1]
[1]Dept. of Electrical and Computer Engineering, Seoul National University, Korea
[2]Memory Business Division, Samsung Electronics, Korea
[1]{heonhwa, suhwan}@snu.ac.kr, [2]hh.cheong@samsung.com

***Abstract*—This work presents a low-power and fast-locking digital 1.6GHz quadrature clock generator (QCG), which mainly consists of a novel ping-pong phase detection (PPD) controller with a pair of latch-based phase detectors. The proposed PPD scheme compares generated clock signals from a digitally controlled delay line (DCDL) with an input clock for fast coarse lock, resulting in a short locking time. Post-layout simulations of an implementation in 28nm CMOS technology suggest that the proposed work can lock within 13 cycles and produce 4-phase 1.6GHz quality output clocks, which supports a data rate of 6.4Gbps. It achieves an RMS jitter of 1.65ps and an effective peak-to-peak jitter of 1.12ps, offers power efficiency of 0.25mW/Gbps, and occupies an area of 0.00247mm².**

Keywords—multiphase clock generator, fast lock, ping-pong phase detection (PPD), digital quadrature clock generator.

I. Introduction

In contemporary high-speed systems-on-a-chips (SoCs) including memory interface, delay-locked loops (DLLs) have been used extensively [1-6] due to its avoidance of jitter accumulation over multiple reference clock cycles [4]. For next-generation high-speed systems, such as GDDR5X/6, multi-phase schemes including quadrature-data-rate (QDR) are demanded [5], especially to achieve a high data rate of 6.4Gbps for memory systems [7].

Among several approaches, digital-based DLLs have become popular due to their short lock-in time, power efficiency and smaller area compared to analog DLLs [2-8] and therefore have been the subject of recent research for power-saving post-DDR4 DLLs [9].

Several new designs of register-controlled digital DLL including [10] for high-speed applications were introduced, but they require long locking time in addition to large area and high power consumption [4]. In other designs [11-12], a time-to-digital converter (TDC) is used to achieve fast locking, but they are still limited by high area and power consumption [4, 6]. Other approaches based on successive-approximation-register (SAR) scheme tackle the limit of the previous work, but its open-loop architecture cannot offer seamless tracking of variations, which might result in critical performance degradation [4].

In this paper, we introduce a digital DLL-based low-power and fast-locking quadrature clock generator (QCG) using a ping-pong phase detector (PPD). The proposed PPD-QCG achieves a fast coarse lock, followed by a fine locking stage, which reduces the total locking time. The PPD shares a digitally controlled delay line (DCDL) with the output of the QCG to minimize area, and it also offers low-power scheme by gating signal switching in the phase detectors during locked state. The proposed QCG generates 4-phase output clocks operating at 1.6GHz, thus supporting QDR data rate of 6.4Gbps for next-generation memory interfaces.

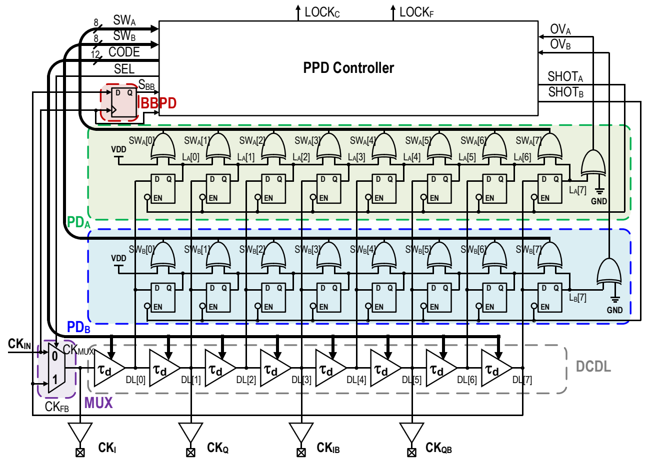

Fig. 1. Block diagram of the proposed PPD-QCG.

II. Architecture

As shown in Fig. 1, the PPD-QCG consists of the following three major blocks: a PPD, a DCDL, and a PPD controller with a multiplexer (MUX) for PPD scheme. The PPD consists of a pair of phase detectors (PD_A and PD_B) for ping-pong style phase detection.

The architecture of each PD is a conventional single-shot phase detector producing one-hot-style output. However, this latch-based PD has its limit on fast phase acquisition while maintaining the previous phase information. A case of contradiction is introduced where an enable input signal of the latches needs to stay low to hold the current value while a high-to-low transition is required to update to a new value.

In this work, we combine the two PDs with a novel PPD control scheme to support seamless acquisition (e.g. multi-shot) of new phase information without losing the previous phase information, which achieves fast locking.

Each PD has an input signal SHOT to capture the phase information (L[0:7]) and outputs an 8-bit signal bus indicating phase information (SW) with an overflow flag (OV) to the PPD controller. The controller detects phase underflow if all bits of SW is zero, while it detects phase overflow based on the OV flag. It then controls the DCDL by CODE signal bus, and also performs DCDL input multiplexing using SEL.

The DCDL consists of 8 equal digitally-controlled unit delay cells (UDCs) of which the delay is τ_d, where each UDC is a combination of current-starving inverter scheme for fine

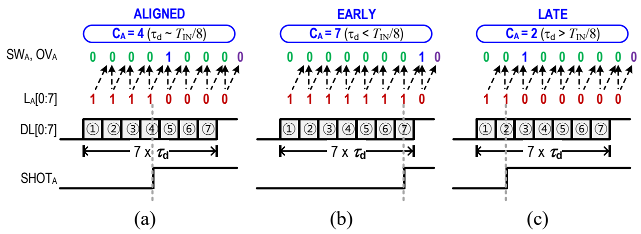

Fig. 2. Phase detection examples of different C_A values for a fixed-phased SHOT signal when the delay of unit delay cell τ_d is (a) similar to the target delay (b) smaller than the target delay (c) greater than the target delay.

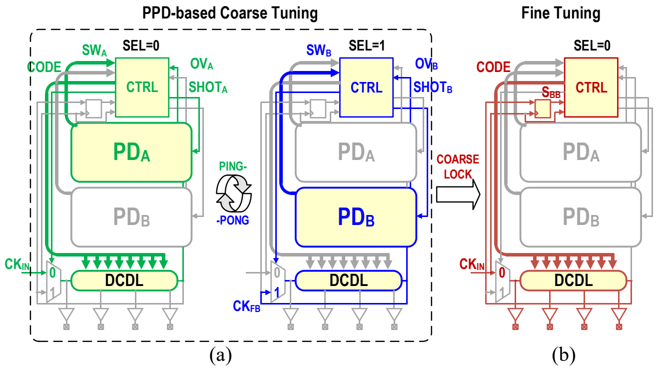

Fig. 3. Simplified diagrams indicating the blocks which are operating during a) the coarse-tuning stage and (b) the fine-tuning stage.

tuning and inverters with shunt-capacitor scheme for coarse tuning, respectively. Total of 8 signals (DL[0:7]) are generated from each UDC, and the 4-phase output clocks (CK_I, CK_Q, CK_{IB}, and CK_{QB}) are then produced.

The BBPD compares the phase between the input clock (CK_{IN}) and a DCDL-generated signal (CK_{FB}), then outputs the result (S_{BB}) to the PPD controller. S_{BB} is then used to determine the sign of next CODE in a fine-tuning stage as well as to finely dither CODE at the final locking stage. The system also outputs two flags $LOCK_C$ and $LOCK_F$ indicating coarse and fine locking, respectively.

III. Fast Locking by the Ping-pong Phase Detection

A. Basic Operations of the Proposed Work

Each PD measures the phase difference between the DL signals and a fixed-phase SHOT signal generated from CK_{IN}. SW signals indicates the relative phase information of DL and SHOT, where the delay of DL is determined by τ_d. The controller then converts SW to a positive integer C, as shown in Fig. 2. This approach not only detects the relative position of generated phase (aligned, early and late), but also obtains the relative magnitude of the phase for fast-locking coarse calibration of DCDL delay performing by the controller.

The ping-pong control scheme initially activates PD_A while it deactivates PD_B at the first cycle of CK_{IN}. In the next cycle of CK_{IN}, the scheme deactivates PD_A while it activates PD_B. This on-and-off cycle for the 2 cycles of CK_{IN} constitutes one PPD cycle. From the third cycle of CK_{IN}, the scheme is repeated until the coarse locking and followed by fine-tuning stage, as depicted in Fig. 3.

B. Controls of Signals for Ping-pong Phase Detection

The PPD controller generates main control signals by the circuit in Fig. 4. First, $SHOT_A$ is generated by a flip-flop capturing of the delayed CK_{IN} for $\tau_r = (T_{IN} / 8) \times N_S$, where

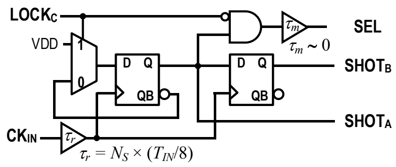

Fig. 4. Schematic of ping-pong control signal generator in PPD controller.

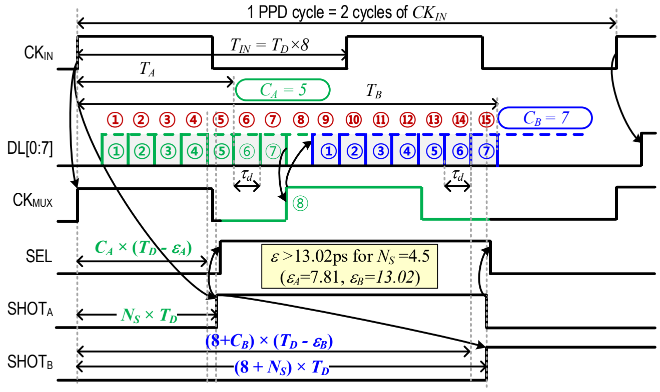

Fig. 5. Simplified timing diagram of ping-pong process for a case of $\tau_d < T_D$.

$1 < N_S < 8$ and T_{IN} is the period of CK_{IN}. $SHOT_B$ is then generated by another flip-flop capturing of $SHOT_A$. SEL is also generated from $SHOT_A$ by delaying it for τ_m, where τ_m is a slight delay allowing time for the phase captures in the PDs before shuffling the multiplexer input. Note that a flag indicating the coarse lock ($LOCK_C$) stops the toggling of control signals so that the system enters a fine-tuning stage. This signal gating scheme also performs switching power reduction in the PDs by stopping phase captures.

C. Error Estimation by the First Shot in Ping-pong Cycle

The initial timing error can be estimated by $SHOT_A$ from the region T_A in Fig. 5. We first define τ_d and T_D, respectively indicating actual and ideal delay value of UDC, as

$$\tau_d = T_D - \varepsilon = (T_{IN} / 8) - \varepsilon \,, \qquad (1)$$

where ε is the delay error of UDC from the ideal value. From T_A, we can derive a formula to estimate timing error ε_A by

$$C_A \times (T_D - \varepsilon_A) < N_S \times T_D \,, \qquad (2)$$

which can be also expressed in terms of ε_A as follows:

$$\varepsilon_A > \frac{C_A - N_S}{C_A} \times T_D \,, \qquad (3)$$

where $C_A \neq floor(N_S)$ with no under-/over-flow occurring. If $C_A = floor(N_S)$, the system adopts another estimation ε_B from the second shot of the PPD cycle, or enters the fine-tuning stage depending on the value of C_B.

D. Error Estimation by the Second Shot in Ping-pong Cycle

Better estimation ε_B can be obtained by $SHOT_B$ from T_B. The error estimation ε_B from T_B can be obtained as

$$(8 + C_B) \times (T_D - \varepsilon_B) < (8 + N_S) \times T_D \,, \qquad (4)$$

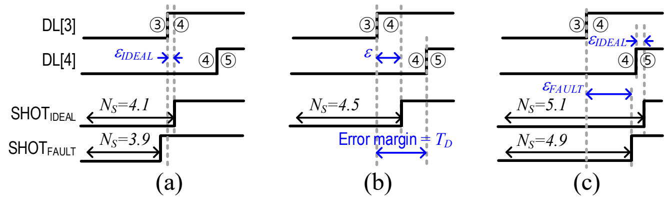

Fig. 6. Examples showing the effect of location of SHOT signal when it is located (a) near the rising edge of DL[3] (b) at the center between the rising edges of DL[3] and DL[4] (c) near the rising edge of DL[4].

TABLE I. NUMERICAL EXAMPLE OF CODE CONTROL

$\#_{PPD}$	N_S	τ_d (ps)	C_A	C_B	$\varepsilon_{A,min}$ (ps)	$\varepsilon_{B,min}$ (ps)	ε (ps)	Δ_τ (ps)
1	4.9(W)	62	5	7	1.6	10.9	10.9	11
2	4.5 (N)	73	4	5	N/A	3.0	3.0	3
3	4.1(B)	76	4	4	Coarse locked		2.4	2

* *Target period (T_D) = 78.125ps (1.6 GHz)*

which can be also expressed as:

$$\varepsilon_B > \frac{C_B - N_S}{C_B + 8} \times T_D . \tag{5}$$

From (3) and (5), the final estimate ε can be found by

$$\varepsilon > \varepsilon_{\min} = \max(\varepsilon_A, \varepsilon_B) . \tag{6}$$

After each PPD cycles, the system updates τ_d based on the estimated error ε by the following:

$$\tau_{d,NEW} = \tau_d + \Delta_\tau \text{ , where} \tag{7}$$

$$\Delta_\tau = round(\varepsilon) . \tag{8}$$

This update is repeated until the following is satisfied:

$$C_A = C_B = floor(N_S) . \tag{9}$$

When (8) is satisfied, the PPD moves to a fine-tuning stage.

E. Delay Line controls of Fine-Tuning Stage

Once the system enters the fine-tuning stage, the system first chooses the initial Δ_τ value by

$$\Delta_\tau = \mathrm{sgn}(S_{BB}) \times floor\left(\frac{0.5 \times T_D}{8} \times 0.5\right), \tag{10}$$

where S_{BB} is the BBPD output indicating τ_d is smaller (+1) or greater (-1) than T_D. From this initial Δ_τ, the controller performs a conventional binary-search tuning based on S_{BB}. This only takes a few cycles because τ_d is already well near the target. After fine locking, only the output-generating DCDL and the code-dithering BBPD are active while other blocks are hibernated or turned off to achieve power saving.

F. Numerical Discussion of the Proposed Work

In this section, a numerical example of the PPD system will be introduced to explain the locking procedure visited in the previous sections. This example is a case of generating

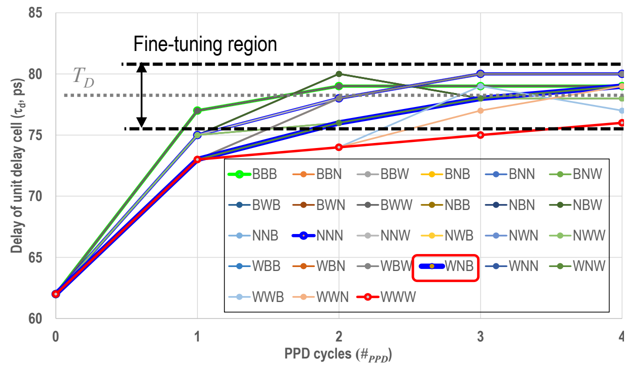

Fig. 7. Transient results of τ_d convergence versus $\#_{PPD}$ achieving coarse locking within 4 PPD cycles for any set of locking strategies. *(For examaple, WNB means a set of Worse-Normal-Better strategies during PPD cycles 1-3)*

1.6GHz (T_D = 78.125ps) output clocks and the delay line is initially off-calibrated at 2.016GHz (τ_d = 62ps).

Precise calibration of N_S right after the rising edge of the DCDL-generated clock ($\mathrm{SHOT_{IDEAL}}$), as depicted in Fig. 6 (a) and (c), offers more accurate ε. It becomes obvious from (3) and (5) that this increases $\varepsilon_{\min}$, offering more aggressive DCDL tuning. However, the non-ideal nature of the circuitry may result in the actual SHOT in the circuit becoming $\mathrm{SHOT_{FAULT}}$ as depicted in Fig. 6. Therefore, aligning the rising edge of SHOT near the center between the two adjacent DLs (e.g. $N_S \simeq 4.5$) would be a more practical approach.

Table I demonstrates the decision procedure of Δ_{CODE} after each PPD cycle ($\#_{PPD}$) for a case where N_S is changing every PPD cycles, in the order of Worse-Normal-Better (N_S: 4.9 → 4.5 → 4.1). The choice N_S for ε estimation can be considered as the PPD locking strategy, and the simulation results in Fig. 7 show the convergence of τ_d within 4 PPD cycles (8 $\mathrm{CK_{IN}}$ cycles) for any set of PPD locking strategy.

Note that T_D and N_S are pre-determined values. In addition, C_A and C_B are bounded positive integers, thus most of the possible values of Δ_τ, rounded values of ε, can be pre-calculated and stored in the system, avoiding the use of any complex arithmetic block while offering low-cost design. In addition, ε_B determines the bound of ε in general, but ε_A is still useful for invalid-ε_B cases due to overflow, as in Fig. 9.

IV. SIMULATION RESULTS

The work was implemented within the area of 0.0025mm^2 in 28nm CMOS process, and the post-layout results with a noise-injected input clock show that it generates 1.6GHz quadrature outputs after 13 input clock cycles, as shown in Figs. 8-10. This work provides the worst-case effective peak-to-peak output jitter (J_{PP}) of 1.12ps and an RMS jitter (J_{RMS}) of 1.65ps while achieving power efficiency of 0.25mW/Gbps. Note that J_{RMS} is mainly from the injected noise of input clock.

Table II summarizes the performance of recent works, and here we have chosen a popular Figure-of-Merits ($\mathrm{FoM_{RMS}}$) for fast-locking clock generators which is a function of locking cycles along with jitter and power. To compare the work without J_{RMS}, we have used $\mathrm{FoM_{PP}}$ which only replaces J_{RMS} in $\mathrm{FoM_{RMS}}$ to J_{PP}. It is shown that work achieves the best $\mathrm{FoM_{PP}}$ and power efficiency (η_p) among all work listed. It also achieves competitive $\mathrm{FoM_{RMS}}$ while still offering the best $\mathrm{FoM_{RMS}}$ among all multi-phase DLLs, considering decent $\mathrm{FoM_{COST}}$ indicating normalized cost across different

TABLE II. COMPARISON OF FAST-LOCKING CLOCK-GENERATING DIGITAL DLLS

Year	2014	2015	2015	2015	2015	2018	2018	2018	2020	This Work [a]
Conf. / Journal	TCAS-II [1]	EL [7]	JSSC [8]	TCAS-I [6]	TVLSI [2]	ISCAS [9][a]	TCAS-I [4]	TCAS-II [13]	TCAS-II [5]	
Process (nm)	130	65	65	55	130	65	130	65	28[b]	28
Supply (V)	1.2	1	1	1	1.5	1	1.2	1	1	1
Area (mm^2)	0.025	0.025	0.0153	0.018	0.08	0.02	0.0077	0.019	0.0072	0.0025
Range (GHz)	0.4-0.8	1.5-5.0	0.003-1.8	0.1-2.5	0.008-0.5	1.65-7.0	1.5-3.3	0.7-2.0	1.8-2.5	0.8-2.0
I-Q support	Yes	No	No	No	Yes	No	No	No	Yes	Yes
Lock cycles	75	11	5	8	8	6	16	40	72	13
J_{PP} / J_{RMS} (ps) @ freq. (Hz)	20 / 2.3 @ 0.8G	6[c] / N/A @ 5.0G	3 / 0.85 @ 1.8G	3 / 0.24 @ 2.5G	10 / 2.3 @ 0.18G	4.55[c] / N/A @ 7.0G	9.3[c] / 1.62 @ 3.3G	4.5[c] / 1.574 @ 2.0G[d]	1.7[c] / 1.05 @ 2.5G	1.12[c] / 1.65 @ 1.6G
Power (mW)	7.200	6.900	9.500	1.960	26.000	7.100	7.000	3.310[e]	3.700	1.623
FoM_{RMS}[f]	-186.7	N/A	-217.7	-231.4	-200.6	N/A	-203.3	-198.8	-196.7	-211.3
FoM_{PP}[g]	-167.9	-195.2	-206.7	-209.5	-187.8	-202.8	-188.1	-189.7	-192.6	-214.6
η_p[h]	2.25	0.69	2.64	0.39	36.11	0.51	1.06	0.83	0.37	0.25
FoM_{COST}[i]	9.25	8.17	19.11	4.67	303.89	4.80	0.67	7.44	13.59	3.20

a. Post-layout simulation results. b. 28nm FDSOI process. c. Effective p-p jitter = output p-p jitter – input p-p jitter [7, 9, 13]. d. Results from 1x-MDLL mode. e. Power at 1GHz.

f. $FoM_{RMS} = 10\log\{(J_{RMS} / 1sec)^2 \times (\text{lock cycles}^2) \times \text{power (mW)}\}$ [14-16]. g. $FoM_{PP} = 10\log\{(J_{PP} / 1sec)^2 \times (\text{lock cycles}^2) \times \text{power (mW)}\}$. h. η_p = power (mW) / data rate (DDR or QDR; Gbps).

i. $FoM_{COST} = FoM_{POWER} \times FoM_{AREA}$, where FoM_{POWER} = power (μW) / (frequency (MHz) × supply^2 (V^2)) and FoM_{AREA} = area (mm^2) / channel length (μm^2) [8, 17, 18].

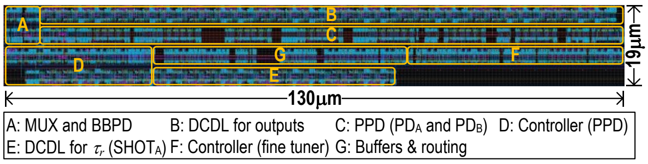

Fig. 8. Layout of the proposed PPD-QCG in 28nm CMOS.

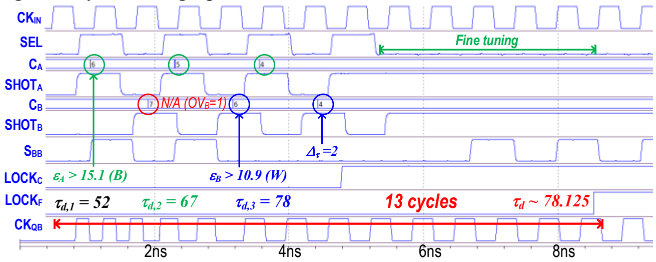

Fig. 9. Post-layout simulation results showing locking of 13 cycles.

process technology nodes and superior power efficiency.

V. CONCLUSION

We have proposed a PPD-QCG, offering novel timing error estimation to achieve fast-locking control of the DCDL as well as dynamic power gating of the PDs. In post-layout simulations, 1.6GHz quadrature clocks were generated within 13 cycles with 0.25mW/GHz power efficiency, which fits the specification of future interface systems requiring 6.4Gbps.

ACKNOWLEDGEMENT

This work was supported by Samsung Electronics.

REFERENCES

[1] K. Ryu et al., "Process-variation-calibrated Multiphase Delay Locked Loop with a Loop-embedded Duty Cycle Corrector," *TCAS-II*, vol. 61, no. 1, pp. 1-5, 2014.

[2] D. Jhang et al., "A Multiphase DLL with a Novel Fast-Locking Fine-Code Time-to-Digital Converter," *TVLSI*, vol. 23, pp. 2680-2684, 2015.

[3] M. Hsieh et al., "A 6.7 MHz to 1.24 GHz 0.0318mm^2 Fast-Locking All-Digital DLL Using Phase-Tracing Delay Unit in 90 nm CMOS," *JSSC*, vol. 51, pp. 412-427, 2016.

[4] E. Bayram et al., "1.5-3.3 GHz, 0.0077mm^2, 7 mW All-Digital Delay-Locked Loop with Dead-Zone Free Phase Detector in 0.13μm CMOS," *TCAS-I*, vol. 65, pp. 39-50, 2018.

[5] Y. Yoon et al., "A DLL-based Quadrature Clock Generator with a 3-stage Quad Delay Unit using the Sub-range Phase Interpolator for Low-jitter and High-phase Accuracy DRAM Applications," *TCAS-II*, vol. 67, pp. 2342-2346, 2020.

[6] J. Wang and C. Cheng, "An All-Digital Delay-Locked Loop Using an In-Time Phase Maintenance Scheme for Low-Jitter Gigahertz Operations," *TCAS-I*, vol. 62, pp. 395-404, 2015.

[7] D. Lee and J. Kim, "5 GHz All-digital Delay-Locked Loop for Future Memory Systems beyond Double Data Rate 4 Synchronous Dynamic Random Access Memory," *Elec. Lett.*, vol. 51, pp. 1963-1975, 2015.

[8] J. Wang et al., "A Wide-Range, Low-Power, All-Digital Delay-Locked Loop with Cyclic Half-Delay-Line," *JSSC*, vol. 50, pp. 2635-2644, 2015.

[9] D. Park and J. Kim, "A 7-GHz Fast-Lock 2-Step TDC-based All-Digital DLL for Post-DDR4 SDRAM," *ISCAS*, 2018.

[10] A. El-Shafie and S. Habib, "A Novel N-bit SAR Implementation for All-digital DLL Circuits," *ICM*, pp. 427-430, 2010.

[11] C. Chung and C. Lee, "A New DLL-based Approach for All-digital Multiphase Clock Generation," *JSSC*, vol. 39, pp. 469-475, 2004.

[12] K. Kim, et al., "A 1.3-mW, 1.6-GHz Digital Delay-Locked Loop with Two-cycle Locking Time and Dither-free Tracking," *SVLSI*, 2013.

[13] J. Kim and S. Han, "A Fast-Locking All-Digital Multiplying DLL for Fractional-Ratio Dynamic Frequency Scaling," *TCAS-II*, vol. 65, pp. 276-280, 2018.

[14] J. Lin and C. Yang, "A Fast-Locking All-Digital Phase-Locked Loop With Dynamic Loop Bandwidth Adjustment," *TCAS-I*, vol. 62, pp. 2411-2422, 2015.

[15] H. Sahu et al., "A Low-jitter Digital-to-time Converter with Look-ahead Multi-phase DDS," *LASCAS*, 2016.

[16] P. Paliwal et al., "A Fast Settling Fractional-N DPLL With Loop-Order Switching," *TVLSI*, vol. 28, pp. 714-725, 2020.

[17] R. Yang and S. Liu, "A 2.5 GHz All-digital Delay-locked Loop in 0.13μm CMOS Technology," *JSSC*, vol. 42, no. 11, pp. 2338-2347, 2007.

[18] C. Cheng et al., "Design of a 2.5-GHz, 3-ps Jitter, 8-locking-cycle, All-digital Delay-locked Loop with Cycle-by-cycle Phase Adjustment," *SVLSI*, 2012.

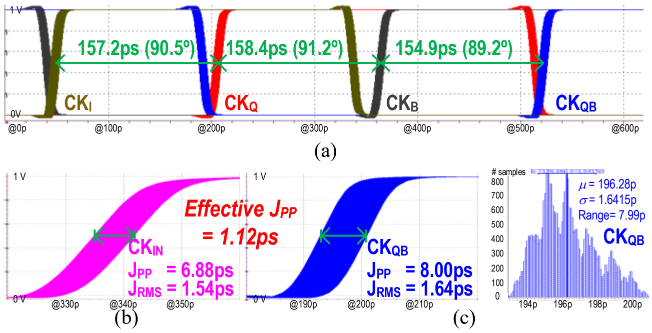

Fig. 10. Post-layout simulation results of (a) generated quadrature clocks, and jitter of (b) noise-injected CK_{IN}, and (c) CK_{QB} output with jitter distribution. *(More than 30,000 samples of the rising egdes were measured)*