A 0.45 pJ/b, 6.4 Gb/s Forwarded-Clock Receiver with DLL-based Self-tracking Loop for Unmatched Memory Interfaces

Soyeong Shin, *Student Member, IEEE*, Han-Gon Ko, *Student Member, IEEE*, Chan-Ho Kye, *Student Member, IEEE*, Sang-Yoon Lee, *Student Member, IEEE*, Jaekwang Yun, Doobock Lee, Hae-Kang Jung, Suhwan Kim, *Senior Member, IEEE*, and Deog-Kyoon Jeong, *Fellow, IEEE*

***Abstract*—This brief presents a power- and area-efficient forwarded-clock (FC) receiver with a delay-locked loop (DLL)-based self-tracking loop for unmatched memory interfaces. In the proposed FC receiver, the self-tracking loop is composed of two-stage cascaded DLLs to support a burst mode. The proposed scheme can compensate for a delay drift without data (DQ) transitions or a re-training by utilizing a *write* training of the memory controller and fixing a data strobe (DQS) path delay through DLLs. The proposed FC receiver is fabricated in the 65-nm CMOS technology and the active area including 4 DQ lanes is 0.0329 mm^2. After the *write* training is completed at supply voltage of 1 V, the measured timing margin remains larger than 0.31 UI when the supply voltage drifts in the range of 0.94 V and 1.06 V from the training voltage, 1 V. At the data rate of 6.4 Gb/s, the proposed FC receiver achieves an energy efficiency of 0.45 pJ/bit.**

***Index Terms*—Delay-locked loop, forwarded-clock receiver, memory interface, timing margin, unmatched type receiver, *write* training.**

I. Introduction

The high-speed memory interfaces are source-synchronous architecture in which a transmitter sends data and clock and a receiver latches data with the clock [1]. If the transmitted clock is center-aligned with the data and the path delay of clock and data is the same, the receiver will have the largest timing margin. As shown in Fig. 1(a), the matched type receiver has a tDQS replica delay cell in the DQ path to match the delay of the DQS path. Since the amount of delay variation caused by a voltage or temperature (VT) drift is the same in each path, the VT drift does not degrade the timing margin in the matched type receiver. However, the receiver design trend of memory interfaces has moved from a matched type to an unmatched type for lower power consumption as the data rate increases. The unmatched type receiver eliminates the tDQS replica cell in the DQ path as shown in Fig. 1(b) to reduce the power consumption. Therefore, it has a different delay between the DQ path and the DQS path. Thus, a memory interface, which adopts the

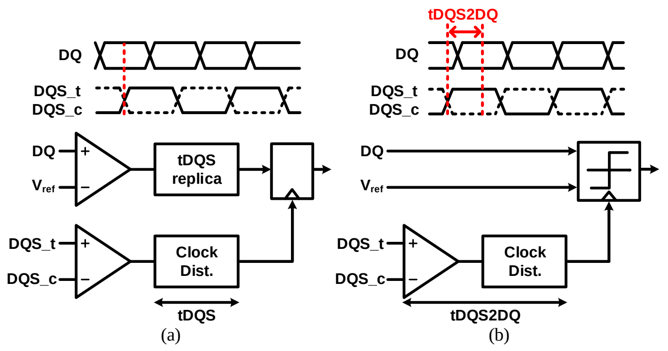


Fig. 1. Type of receiver: (a) matched type (b) unmatched type.

unmatched type receiver, performs a *write* training to locate a DQS transition on the DQ center at the DQ sampler for the optimal timing margin [2]. In the *write* training, after a DQS path delay is measured, a memory controller trasmits DQ later than DQS by the measured DQS path delay, tDQS2DQ. However, the VT drift occuring after the *write* training changes tDQS2DQ and results in the reduced timing margin because the sampling point deviates from the trained sampling point, the DQ center. The reduced timing margin is a critical problem since a DQ eye width is decreased as the data rate increases. Therefore, compensation for the delay drift is required to maintain the timing margin.

To detect the DQS path delay drift, a periodic incremental training [3] and an internal DQS clock-tree oscillator [4] are suggested. In [3], the delay variation is monitored by measuring the shift of the DQ eye edge in each refresh cycle. During the refresh, the memory controller transmits '1010' pattern to DQ and errors are counted by sweeping the DQS delay. From the counted number of the errors, the drift of the DQ eye edge can be tracked. In [4], the DQS path delay is traced by an internal DQS clock-tree oscillator which replicates the DQS path delay. To measure the amount of the delay, the counter value of the DQS oscillator is stored in a register during the given time interval in the *write* training. Then, the entire process is repeated after the *write* training to observe the drift of the DQS path

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment.

S. Shin, H.-G. Ko, C.-H. Kye, S.-Y. Lee, J. Yun, S. Kim, and D.-K. Jeong are with the Department of Electrical and Computer Engineering and the Inter-University Semiconductor Research Center, Seoul National University, Seoul 08826, South Korea (e-mail: syshin@isdl.snu.ac.kr, dkjeong@snu.ac.kr).

D. Lee and H.-K. Jung are with the SK Hynix, Icheon 17336, South Korea.

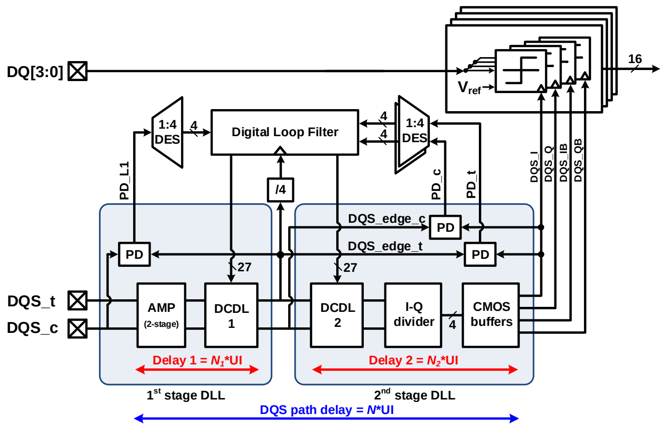


Fig. 2. Overall architecture of the proposed forwarded-clock receiver.

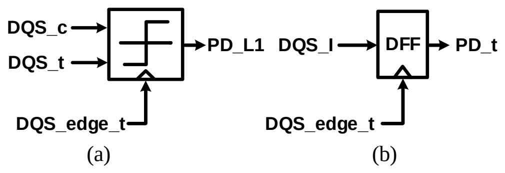


Fig. 3. Phase detector architecture (a) in the 1st stage DLL (b) in the 2nd stage DLL.

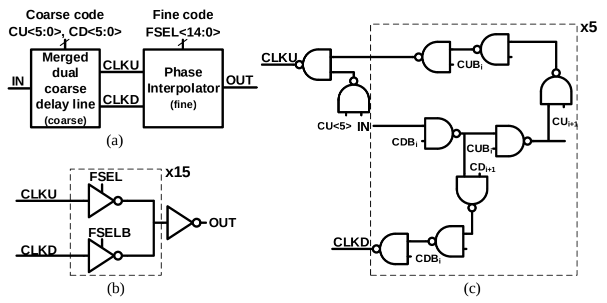


Fig. 4. (a) DCDL architecture (b) phase interpolator (c) merged dual coarse delay line.

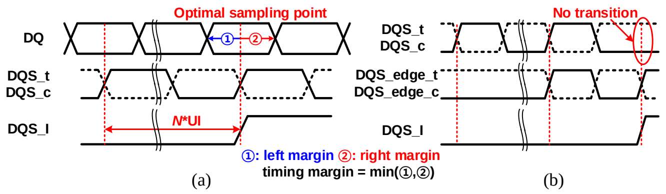


Fig. 5. Timing diagram of the proposed scheme (a) in a seamless mode and (b) in a burst mode.

delay by comparing the counter values. If a large delay drift is detected by using these methods, the memory controller is required to perform a re-training to change the delay setting, which indicates the relative timing relationship between DQ and DQS from the transmitter. Consequently, the memory controller is responsible for the delay drift compensation which incurs significant design complexity.

Instead of implementing a re-training, embedding a DLL in the DQS path can be a feasible solution for self-tracking because the DLL can fix tDQS2DQ by controlling the delay lines even if the VT drift occurs. Moreover, in the *write* training, the relative delay between DQ and DQS is set to tDQS2DQ. As a result, the DQS transition is always centered at the DQ eye and the timing margin is not decreased by the drift. In this work, a power- and area-efficient FC receiver with a DLL-based self-tracking loop is presented, which exploits the *write* training and does not require re-training or DQ transitions for the delay drift compensation.

This brief is organized as follows. Section II details the architecture and the operation of the proposed FC receiver with a DLL-based self-tracking loop. Section III shows the measurement results of the proposed scheme and Section IV concludes the brief.

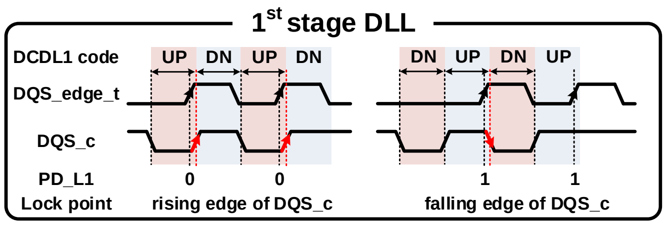
II. Proposed Forwarded Clock Receiver

A. Architecture

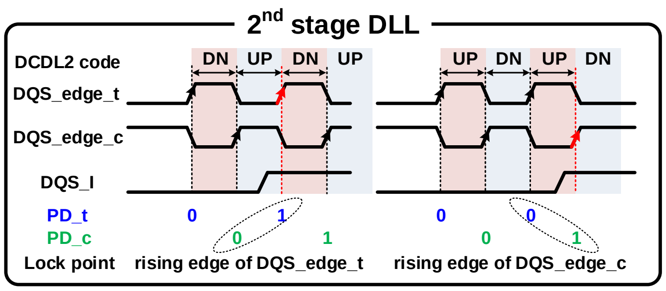
Fig. 2 shows the overall architecture of the proposed FC receiver. The FC receiver consists of a digital loop filter (DLF), 1:4 deserializers (DES), a divider, DQ samplers, and two-stage cascaded DLLs which include phase detectors (PD) and digitally-controlled delay lines (DCDL). Since input DQS (DQS_t and DQS_c) have a low voltage swing from 0 V to 0.4 V, a PMOS input strong-arm latch based sampler is used as the PD of the first-stage DLL in Fig. 3(a). On the other hand, as shown in Fig. 3(b), a D-flip-flop is employed as the PD of the second-stage DLL since DQS_edge_t and DQS_I are full swing signals. The DQS path includes a two-stage amplifier, two DCDLs, an I-Q divider, and CMOS buffers. The two-stage amplifier amplifies input DQS from the 0.4-V peak-to-peak swing to the full rail-to-rail swing. As shown in Fig. 4, DCDL is composed of dual coarse delay lines and a tri-state inverter-based phase interpolator (PI) [5] to cover a wide delay range with fine resolution while avoiding a boundary switching problem. In the coarse delay line, 6-bit thermometer code CU and CD controls the number of on-state NANDs in CLKU and CLKD path respectively, which has a 2*t_{NAND} time difference. The PI interpolates CLKU and CLKD through adjusting the relative strength by the number of the on-state tri-state inverters in each path. The length of the DCDL codes is determined by the required delay range and the DCDL has an effective resolution, 2*t_{NAND}/15. The I-Q divider generates 4-phase clocks for a quarter-rate clocking. Both DCDL1 and DCDL2 are adjusted to fix the total DQS path delay as N times the unit interval (UI) by two-stage cascaded DLLs.

B. Operation

Fig. 5 shows the timing diagram of the proposed scheme. In Fig. 5(a), DQS_I is N*UI delayed from the input DQS by means of cascaded DLLs. Since the optimal sampling point is set in the *write* training when tDQS2DQ is N*UI and the cascaded DLLs always fix the DQS path delay to N*UI by adjusting DCDLs, the sampling point is not changed even if the VT drifts. Therefore, the timing margin, which is defined as the minimum value of the left margin and the right margin from the DQ center, can be kept constant by the DLLs even though the VT drift occurs. As shown in Fig. 5(b), in the burst mode and long tDQS2DQ condition, phase comparison between the input DQS and DQS_I cannot be carried out because input DQS does not toggle when DQS_I transition occurs. Thus, to support the burst mode, DQS_edge_t and DQS_edge_c are generated for the phase comparison in the first-stage DLL. In other words, the DLLs are separated into two parts. The first-stage DLL makes DQS_edge and the second-stage DLL generates the DQ



(a)



(b)

Fig. 6. Lock point setting method (a) 1st stage DLL and (b) 2nd stage DLL.

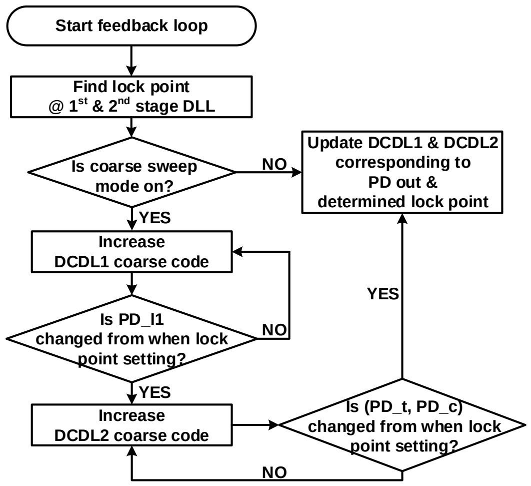

Start feedback loop
Find lock point @ 1st & 2nd stage DLL
Is coarse sweep mode on?
NO
Update DCDL1 & DCDL2 corresponding to PD out & determined lock point
YES
Increase DCDL1 coarse code
Is PD_l1 changed from when lock point setting?
NO
YES
YES
Increase DCDL2 coarse code
Is (PD_t, PD_c) changed from when lock point setting?
NO

Fig. 7. A flow chart of the digital loop filter operation.

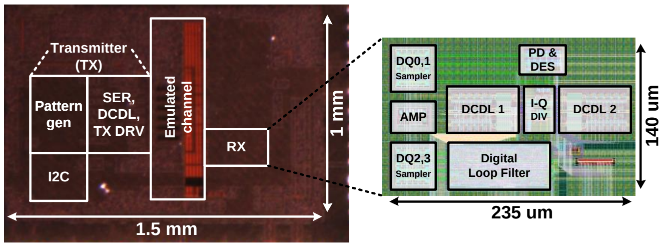


Fig. 8. Chip microphotograph and core layout of the proposed scheme.

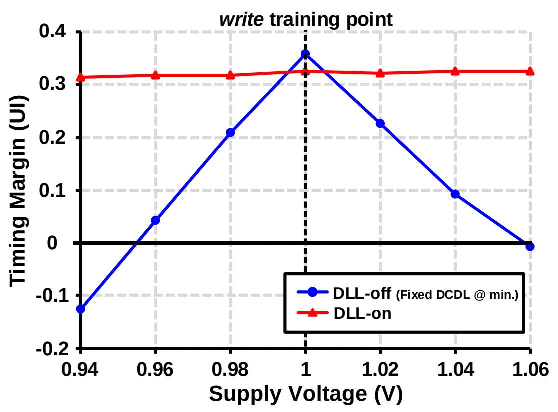


Fig. 9. Measured timing margin vs. supply voltage variation (PRBS7, BER=10^{-9}).

sampling signal, which is N*UI delayed from the input DQS in the locked state.

The first-stage DLL creates DQS_edge, which are N_1*UI delayed version of input DQS, by edge-aligning DQS_edge with input DQS through controlling DCDL1. Since DQS_edge is produced by delaying input DQS, the amount of the delay should be minimized for low jitter condition at the locked state. Thus, the first-stage DLL is implemented to align the rising transition of DQS_edge_t with the transition edge of DQS_c which is the closest to the rising transition of DQS_edge_t and is also later than the rising transition of DQS_edge_t when the codes of the coarse delay line of DCDL1 are at the minimum. Fig. 6(a) shows the lock point setting method of the first-stage DLL. Through phase comparison of DQS_edge_t and DQS_c, PD_L1 calculates the lock point of the first-stage DLL under the minimum coarse delay condition of DCDL1. The determined lock point is the nearest transition edge of DQS_c that can be aligned with the rising edge of DQS_edge_t by the delay increment of DCDL1. For instance, if PD_L1 is ‘0’, the rising edge of DQS_edge_t leads the rising edge of DQS_c and it should be aligned with the rising edge of DQS_c by increasing the DCDL1 delay. Otherwise, if PD_L1 is ‘1’, the rising edge of DQS_edge_t should be aligned with the falling edge of DQS_c. The decided lock point can be varied with the initial delay condition or PVT due to the different rising transition position of DQS_edge_t. Similarly, the lock point of the second-stage DLL is determined as the rising transition of DQS_edge which is the closest to the rising transition of DQS_I when the codes of the coarse delay line of DCDL2 are at the minimum for the same reason and can be aligned with DQS_I by increasing delay of DCDL2. The combination of PD outputs, PD_t and PD_c, indicates the position of the rising transition of DQS_I. As shown in Fig. 6(b), in the case shown on the left, the rising edge of DQS_I is between the rising edge of DQS_edge_c and DQS_edge_t. Thus, the delay of DCDL2 should be increased to make the rising edge of DQS_I aligned with the rising edge of DQS_edge_t. In the case shown on the right, the rising edge of DQS_I can be aligned with the rising edge of DQS_edge_c by increasing the delay of DCDL2. As a result, the DQS path delay from the input DQS to DQS_I is always N*UI at the locked state. If the duty cycle of input DQS is not 50%, the DQS path delay will not be the same as N*UI. However, only the fixed DQS path delay is important because this sampling point will be the optimal sampling point when the memory controller performs the *write* training under this condition and DQS path delay is fixed by controlling DCDLs.

Fig. 7 shows a flow chart of the DLF operation. A coarse sweep mode and an update gain of each DCDL can be selected and controlled respectively by I^{2}C for adjusting the locking time of each DLL. First, the lock point of each stage DLL is determined by the PD output under the initial condition that delay codes of both DCDL1 and DCDL2 are at the minimum. Next, if the coarse sweep mode is on, the coarse codes of each DCDL are increased until the PD output pattern changes from

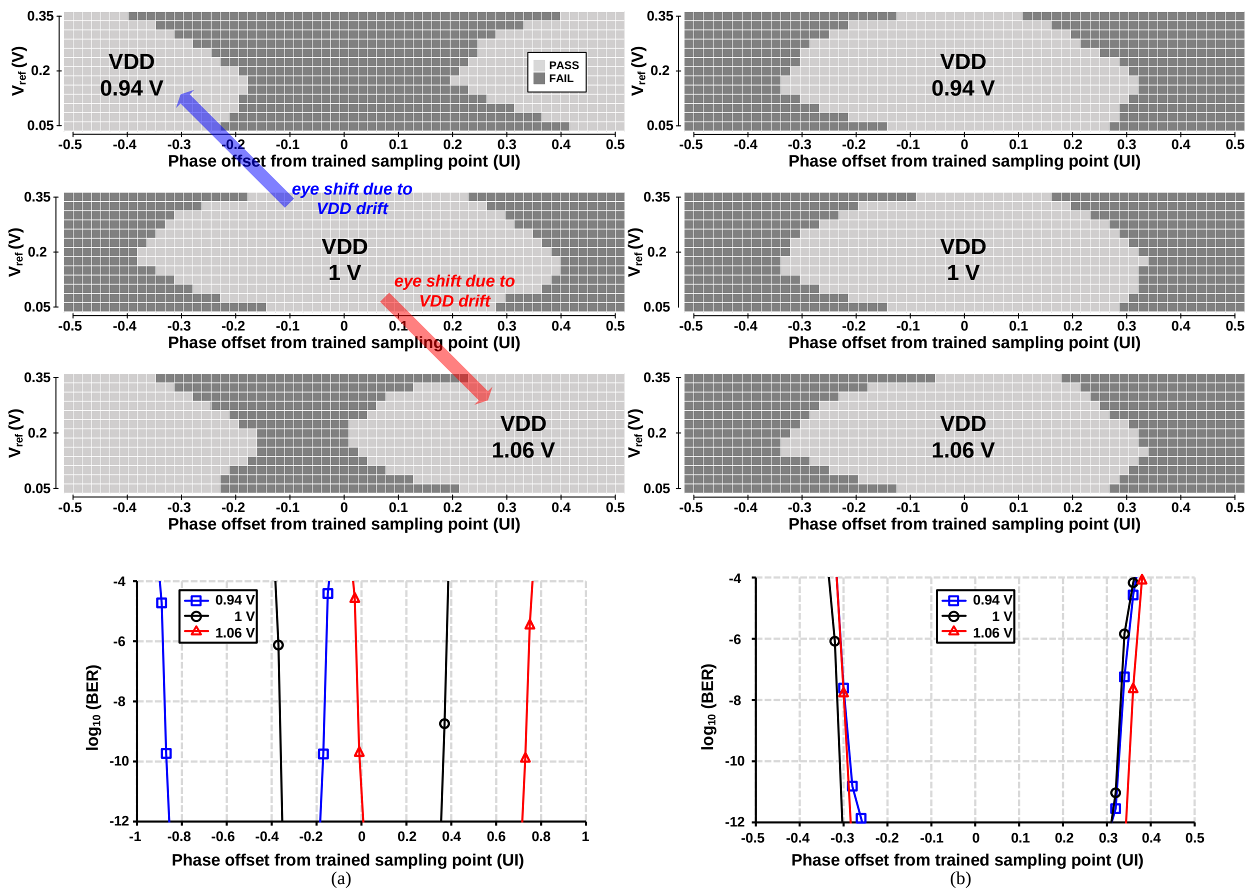
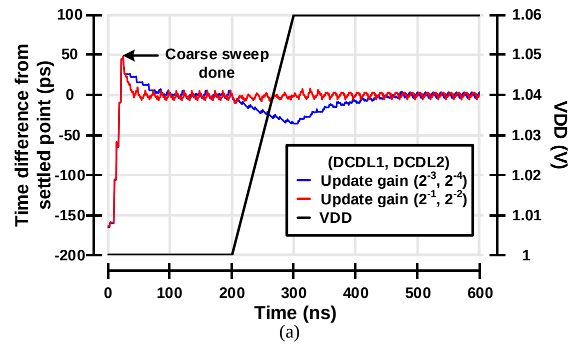


Fig. 10. Measured DQ eye diagram (PRBS7, BER=10^{-9}) and bathtub curve at 0.2-V V_{ref} (a) DLL-off and (b) DLL-on.



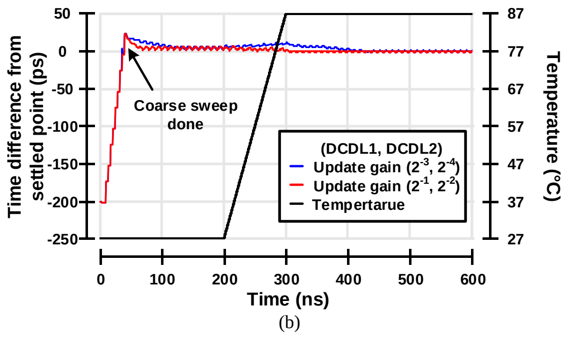


Fig. 11. Simulated tracking behavior of the proposed FC receiver (a) when voltage drift occurs and (b) when temperature drift occurs.

the initial value which is calculated at the lock point setting step. For example, if the PD_L1 is ‘0’ in the lock point setting when the coarse delay of the DCDL1 is at the minimum, the PD_L1

TABLE I. MEASURED POWER BREAKDOWN

Block	Power	Block	Power
DCDL	3.35 mW	DQS path	3.05 mW
Sampler	3.14 mW	DES	0.82 mW
DLF	0.65 mW	PD	0.42 mW
Total	11.45 mW at 6.4 Gb/s, 1-V supply, 4 DQ		

can be changed to ‘1’ by increasing the coarse delay of DCDL1 which implies the rising edge of DQS_edge_t lags behind the rising edge of DQS_c. This means the coarse sweep of the first-stage DLL is done and the following step is the coarse sweep of the second-stage DLL. The coarse sweep of the second-stage DLL is done in the similar way as the first-stage DLL case except for adjusting the delay of DCDL2 instead DCDL1. Then, the codes of DCDL1 and DCDL2 are updated simultaneously corresponding to the PD output and the lock point setting as shown in Fig. 6 because lock point determines which edge to be locked and PD outputs informs the current position of the rising edge of signals to be aligned. In Fig. 6(a), if the lock point of first-stage DLL is the rising edge of DQS_c and PD_L1 indicates that the rising edge of DQS_edge_t is in the UP region, the DCDL1 code is increased to align the risinig edge of DQS_edge_t to the lock point. Otherwise, if PD_L1 is ‘1’ and the rising edge of DQS_edge_t is in DN region, the DCDL1 code is decreased. Also, the DCDL2 code is controlled by the location of the DQS_I rising transition and the lock point of the second-stage DLL as shown in Fig. 6(b).

TABLE II. PERFORMANCE COMPARISON

	ISSCC'12 [7]	CICC'13 [8]	SOVC'14 [9]	TCAS2'16 [10]	TCAS1'16 [11]	SOVC'19 [6]	This work
Technology	90 nm	28 nm	65 nm	65 nm	65 nm	65 nm	65 nm
Architecture	Digital DLL	Digital DLL	PI	ILO + PI	Analog DLL	Analog PD+ Digital DLL	Digital DLL based self-tracking loop
Data rate (Gb/s)	8	6.4	14	10	12.5	4.8	6.4
Clock rate (GHz)	2	3.2	3.5	5	6.25	1.2	1.6
Power (mW)	10.4*	6.71*	7.84	7.1	4.5	7.04**	11.45**
VDD (V)	1.25	0.85	0.8	1	0.9	1.1	1
Power efficiency (pJ/b)	1.3	1.05	0.56	0.71	0.36	0.37	0.45
Area (mm2)***	0.225	0.02	0.36	0.014	0.025	0.0056	0.008
Require data transition ?	O	O	O	O	O	O	X (*Write* training)

*Excludes equalizer power **includes 4 data lanes ***area per lane

III. Measurement Results

The proposed FC receiver is fabricated in the 65-nm CMOS technology. The chip microphotograph and the core layout of the FC receiver are shown in Fig. 8. The DCDL in a transmitter (TX) and the emulated channel are implemented to behave in place of the memory controller and the memory channel, respectively [6]. Using standard cells, the DLF is fully synthesized and automatically placed and routed. The active area of the proposed FC receiver is 0.0329 mm^2, which includes 4 DQ lanes.

The graph of timing margin versus supply voltage is shown in Fig. 9. The DLL-off case is explored when all RX domain DCDL codes are set at the minimum. At 1 V in the DLL-on case, the timing margin is reduced from 0.36 UI to 0.33 UI due to the increased DQS path delay compared with the DLL-off case. However, the timing margin of the DLL-on case remains larger than 0.31 UI while the supply voltage drifts in the range of 0.94 V and 1.06 V after the *write* training is done at 1 V.

Fig. 10 shows the measured DQ eye diagram and the bathtub curve at three different supply voltages when the *write* training is carried out at 1 V. Contrary to the DLL-off case, the DQ eye does not shift because the sampling point is not modified in the DLL-on case.

The simulated tracking behavior of the proposed scheme when the 0.06-V supply voltage drift occurs and 60-℃ temperature drift occurs from 200 ns to 300 ns is shown in Fig. 11. When the DCDL update gain is high, the sampling time difference from the settled point remains constant due to a high loop bandwidth.

Table I shows the measured power breakdown of the proposed FC receiver. The total power consumption is 11.45 mW from 1-V supply at 6.4 Gb/s including 4 DQ lanes. In Table II, the performance of the proposed FC receiver is summarized and compared with the state-of-the-art FC receivers [6]-[11]. The proposed FC receiver uses a small area and low power. Furthermore, the FC receiver is able to operate in the absence of data transitions since the sampling point is fixed by the DLL using only DQS and the *write* training sets the sampling point as the optimal sampling point for the DQ eye centering.

IV. Conclusion

A small-area and power-efficient FC receiver with DLL-based self-tracking loop for unmatched memory interfaces is proposed and implemented in the 65-nm CMOS technology. The proposed FC receiver adopts a cascaded DLL architecture to support the burst mode. The proposed scheme compensates for the VT drift by fixing tDQS2DQ using a DLL. Therefore, it does not require a re-training in the memory controller. Since it utilizes the *write* training, it does not need DQ transitions. In addition, it does not increase the capacitance at DQs because monitoring DQ is not necessary for VT drift compensation. The FC receiver achieves the timing margin larger than 0.31 UI with power efficiency of 0.45 pJ/bit at 6.4 Gb/s while the supply voltage drifts in the range 0.94 V and 1.06 V.

Acknowledgment

This work was supported by SK Hynix.

References

[1] S.-K. Lee *et al.*, "A QDR-Based 6-GB/s Parallel Transceiver With Current-Regulated Voltage-Mode Output Driver and Byte CDR for Memory Interface," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 2, pp. 91-95, Feb. 2013.

[2] LPDDR4 Specification (JESD209-4), JEDEC standard, JEDEC solid state technology association, Aug. 2014.

[3] C. P. Mozak and J. A. Mccall, "Periodic training for unmatched signal receiver," U.S. Patent 9 218 575, Dec. 22, 2015.

[4] C. P. Mozak, "Timing control for unmatched signal receiver," U.S Patent 9 658 642, May 23, 2017.

[5] C. Kim *et al.*, *High-Bandwidth Memory Interface*, NewYork, NY, USA: Springer, 2014, pp. 34-37.

[6] H.-G. Ko *et al.*, "A 370-fJ/b, 0.0056 mm^2/DQ, 4.8-Gb/s DQ Receiver for HBM3 with a Baud-Rate Self-Tracking Loop," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2019, pp. C94-C95.

[7] Y.-S. Kim *et al.*, "An 8GB/s Quad-Skew-Cancelling Parallel Transceiver in 90nm CMOS for High-Speed DRAM Interface," in *Proc. IEEE ISSCC Dig. Tech. Papers*, Feb. 2012, pp. 136-137.

[8] S. Chen *et al.*, "A 1.2 pJ/b 6.4 Gb/s 8+1-Lane Forwarded-Clock Receiver with PVT-Variation-Tolerant All-Digital Clock and Data Recovery in 28nm CMOS," in *Proc. IEEE Custom Integr. Circuits Conf.,* Sep. 2013.

[9] H. Li *et al.*, "A 0.8V, 560fJ/bit, 14Gb/s Injection-Locked Receiver with Input Duty-Cycle Distortion Tolerable Edge-Rotating 5/4X Sub-Rate CDR in 65nm CMOS," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2014.

[10] S.-H. Chung *et al.*, "A 10-Gb/s 0.71-pJ/bit Forwarded-Clock Receiver Tolerant to High-Frequency Jitter in 65-nm CMOS," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 3, pp. 264-268, Mar. 2016.

[11] W. Bae *et al.*, "A 0.36 pJ/bit, 0.025 mm2, 12.5 Gb/s Forwarded-Clock Receiver With a Stuck-Free Delay-Locked Loop and a Half-Bit Delay Line in 65-nm CMOS Technology," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 9, pp. 1393-1403, Sep. 2016.