Ph.D.Dissertation

Design of Voltage and Temperature-tolerant Clock Path and Phase Error Corrector for High-Speed DRAM Interface

고속 DRAM 인터페이스를 위한 전압 및 온도에 둔감한 클록 패스와 위상 오류 교정기 설계

by

Soyeong Shin

February, 2021

Department of Electrical and Computer Engineering
College of Engineering
Seoul National University

Design of Voltage and Temperature-tolerant Clock Path and Phase Error Corrector for High-Speed DRAM Interface

지도 교수 정 덕 균

이 논문을 공학박사 학위논문으로 제출함
2021 년 2 월

서울대학교 대학원
전기·정보공학부
신 소 영

신소영의 박사 학위논문을 인준함
2021 년 2 월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Design of Voltage and Temperature-tolerant Clock Path and Phase Error Corrector for High-Speed DRAM Interface

by

Soyeong Shin

A Dissertation Submitted to the Department of
Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
at
SEOUL NATIONAL UNIVERSITY

February, 2021

Committee in Charge:

Professor Dongsuk Jeon, Chairman

Professor Deog-Kyoon Jeong, Vice-Chairman

Doctor Hankyu Chi

Professor Woo-Seok Choi

Professor Kang Yoon Lee

Abstract

Design of Voltage and Temperature-tolerant Clock Path and Phase Error Corrector for High-Speed DRAM Interface

Soyeong Shin
Department of Electrical and Computer Engineering
The Graduate School
Seoul National University

To cope with problems caused by the high-speed operation of the dynamic random access memory (DRAM) interface, several approaches are proposed that are focused on the clock path of the DRAM. Two delay-locked loop (DLL) based schemes, a forwarded-clock (FC) receiver (RX) with self-tracking loop and a quadrature error corrector, are proposed. Moreover, an open-loop based scheme is presented for drift compensation in the clock distribution. The open-loop scheme consumes less power consumption and reduces design complexity.

The FC RX uses DLLs to compensate for voltage and temperature (VT) drift in unmatched memory interfaces. The self-tracking loop consists of two-stage cascaded DLLs to operate in a DRAM environment. With the write training and the proposed DLL, the timing relationship between the data and the sampling clock is always op-

timal. The proposed scheme compensates for delay drift without relying on data transitions or re-training. The proposed FC RX is fabricated in 65-nm CMOS process and has an active area containing 4 data lanes of 0.0329 mm^2. After the write training is completed at the supply voltage of 1 V, the measured timing margin remains larger than 0.31-unit interval (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 RX achieves an energy efficiency of 0.45 pJ/bit.

Contrary to the aforementioned scheme, an open-loop-based voltage drift compensation method is proposed to minimize power consumption and occupied area. The overall clock distribution is composed of a current mode logic (CML) path and a CMOS path. In the proposed scheme, the architecture of the CML-to-CMOS converter (C2C) and the inverter is changed to compensate for supply voltage drift. The bias generator provides bias voltages to the C2C and inverters according to supply voltage for delay adjustment. The proposed clock tree is fabricated in 40 nm CMOS process and the active area is 0.004 mm^2. When the supply voltage is modulated by a sinusoidal wave with 1 MHz, 100 mV peak-to-peak swing from the center of 1.1 V, applying the proposed scheme reduces the measured root-mean-square (RMS) jitter from 3.77 ps_{RMS} to 1.61 ps_{RMS}. At 6 GHz output clock, the power consumption of the proposed scheme is 11.02 mW.

A DLL-based quadrature error corrector (QEC) with a wide correction range is proposed for the DRAM whose clocks are distributed over several millimeters. The quadrature error is corrected by adjusting delay lines using information from the phase error detector. The proposed error correction method minimizes increased jitter due to phase error correction by setting at least one of the delay lines in the quad-

rature clock path to the minimum delay. In addition, the asynchronous calibration on-off scheme reduces power consumption after calibration is complete. The proposed QEC is fabricated in 40 nm CMOS process and has an active area of 0.048 mm^2. The proposed QEC exhibits a wide correctable error range of 101.6 ps and the remaining phase errors are less than 2.18° from 0.8 GHz to 2.3 GHz clock. At 2.3 GHz, the QEC contributes 0.53 ps_{RMS} jitter. Also, at 2.3 GHz, the power consumption is reduced from 8.89 mW to 3.39 mW when the calibration is off.

Keywords : Clock tree, delay-locked loop (DLL), dynamic random access memory (DRAM) interface, forwarded clock receiver (FC RX), open-loop drift compensation, phase error, quadrature error corrector (QEC), temperature drift, timing margin, unmatched type receiver, voltage drift, voltage noise, write training.

Student Number : 2016-20926

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As shown in Figure 1.1 [1], the bandwidth of dynamic random access memory (DRAM) increases as the demand for high speed increases due to machine learning, big data processing, etc. To cope with the demand, quadrature clocking and the unmatched receiver are used as mentioned in Section 2.2 and Section 4.1 respectively.

To maximize the timing margin of the sampler in the DRAM receiver, the sampling clock should be located at the center of the data eye. Therefore, the memory controller unit (MCU) transmits data (DQ) and data strobe (DQS) with the calculated timing relationships for optimal timing margin. However, the DQS path and the DQ path are different in the unmatched DQ receiver. Although the write training procedure can locate the DQS at the DQ center, the timing margin is reduced by

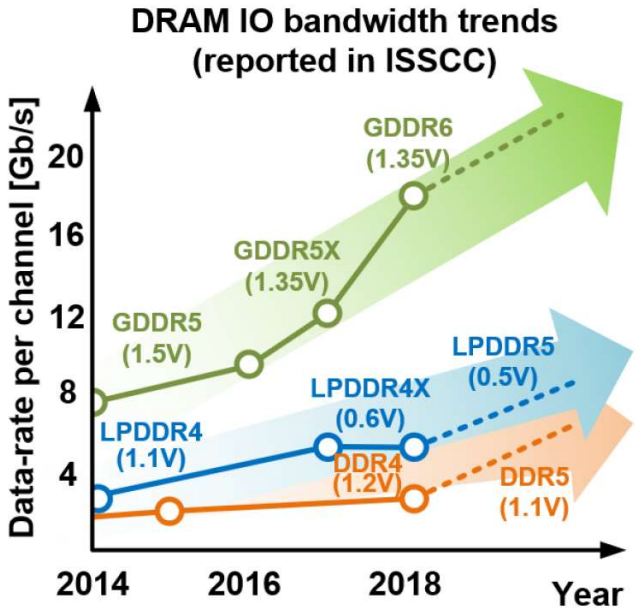


Fig. 1.1 DRAM bandwidth trends [1].

voltage or temperature (VT) variation. The DQS interval oscillator and periodic training are proposed to solve the problem [2], [3]. However, these solutions require additional time for retraining, thus reducing bandwidth. A DLL-based self-tracking loop for VT drift compensation is proposed for background calibration [4]. Since DQS only toggles when DQ is transmitted, the DLL of the proposed scheme to operate in the DRAM interface is separated into two DLLs.

The clock distribution of DRAM consists of global distribution and local network. The global clock is distributed as a current mode logic (CML) level and converted to full swing in the local network to transmit and receive data. Similar to the delay drift in the receiver, the supply voltage fluctuation deteriorates the noise of the distributed clock, degrading the quality of data synchronized with the clock. The closed-loop

method such as DLL can compensate for the drift, but it occupies an area and consumes more power than the open-loop scheme. The open-loop based methods are suggested [5], [6]. However, these schemes decrease bandwidth due to the increased capacitance intentionally or require a bandgap reference which causes design complexity. In this thesis, the proposed voltage drift compensation scheme is implemented with a simple architecture, current starved inverters and a bias generator. The bias generator generates bias voltages to control the current in the inverter and control a feedback resistor in the CML-to-CMOS converter to compensate for drift.

In the quadrature clocking of the DRAM transmitter, phase error between quadrature clocks reduces the eye width of the transmitted data. Therefore, accurate multi-phase generators and quadrature error correctors are suggested [7]-[11]. However, these methods require an oscillator or degrade the quality of the corrected clocks due to the increased delay in the clock path. A digital DLL-based quadrature error corrector is proposed that minimizes the contributed jitter of the phase corrector [12]. The proposed scheme minimizes the increased delay due to phase error correction, so the increased jitter of output clocks of the quadrature error corrector can be minimized. Also, an asynchronous calibration on-off scheme is implemented to reduce power consumption in the idle state.

1.2 Thesis Organization

This thesis is organized as follows. In Chapter 2, the background of the DRAM interface is explained. The basic operation of the transmitter and the receiver in DRAM is described. Also, the reason why VT drift compensation and accurate phase difference, which are the motives of this thesis, are important is mentioned.

In Chapter 3, the background of the DLL is explained. The basic operation and the building blocks of the DLL are presented.

In Chapter 4, a forwarded-clock receiver with a DLL-based self-tracking loop for unmatched memory interfaces is described. To operate in a DRAM environment, the DLL is separated into two loops and the operation of the proposed DLL is explained. Then, the circuit implementation is described and the measurement results are shown.

In chapter 5, an open-loop-based voltage drift compensation in the clock distribution is presented. Prior works are shown and the open-loop voltage drift compensation method is explained. Then, the circuit implementation is explained and the measurement results are shown.

In chapter 6, a quadrature error corrector is presented. Prior works are shown and the quadrature error correction method which minimizes the increased delay is explained. Also, an asynchronous calibration on-off scheme is presented. Then, the circuit implementation is explained and the measurement results are shown.

Chapter 7 summarizes the proposed works and concludes this thesis.

Chapter 2

Background on DRAM Interface

2.1 Overview

Synchronous dynamic random access memory (SDRAM) is used widely for a smartphone, personal computer, tablet, and so on. In SDRAM, a command or data is synchronized to an external clock for processing high-speed data. For example, READ or WRITE command is synchronous with CK and data (DQ) is synchronous with data strobe (DQS). There are two types of SDRAM, single data rate (SDR) and double data rate (DDR), as shown in Figure 2.1 [13]. In SDR, DQ is synchronized to the rising edge of CLK like full-rate system. Contrary to SDR, DQ is synchronized to both the rising edge and the falling edge of CLK like half-rate system. Also, DDR can be categorized by its application. The graphics memory GDDR is used for gaming, processing parallel data, and has a high per-pin data rate. The main memory

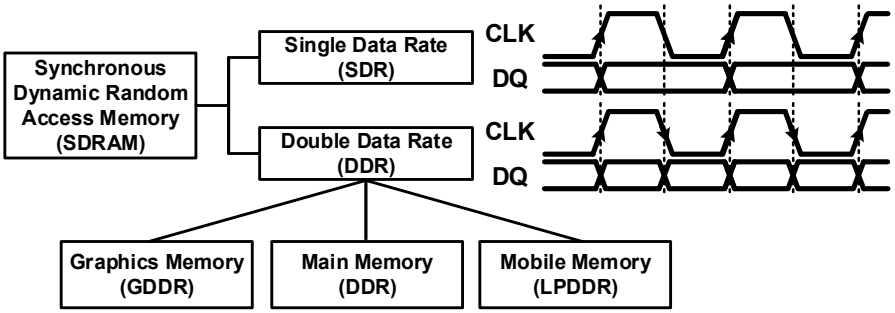


Fig. 2.1 Types of DRAM.

Fig. 2.2 Communication between MCU and SDRAM.

DDR, which is also known as RAM, is used in servers and PC. The mobile memory LPDDR, which concentrates on the low power consumption, is used for a mobile phone and tablet.

The memory controller unit (MCU) and SDRAM are communicated with each other as shown in Figure 2.2 [13]. The MCU transmits CLK and commands such as READ or WRITE to SDRAM. Afterward, SDRAM sends the data stored in the DRAM cell when the READ command is received. When a WRITE command is received, SDRAM receives the data from the MCU and stores the received data in the DRAM cell.

2.2 Memory Interface

Figure 2.3 shows the block diagram of DRAM. The banks are at the top and bottom of the DRAM. The bank includes the cell capacitor, bit-line sense amplifier, row path, column path, and so on. The row decoder in the row path selects the target word line. The column decoder in the column path chooses the target bit line. As a result, the target DRAM cell is selected and the bit-line sense amplifier amplifies the voltage difference between the reference voltage and the selected bit line. The peripheral circuits are in the center of the DRAM.

The peripheral circuit includes a clock tree, a DQ transmitter (TX), a DQ receiver (RX), a delayed-lock loop (DLL), etc. In Figure 2.4, the global clock buffer distributes clocks to several DQs to send and receive data.

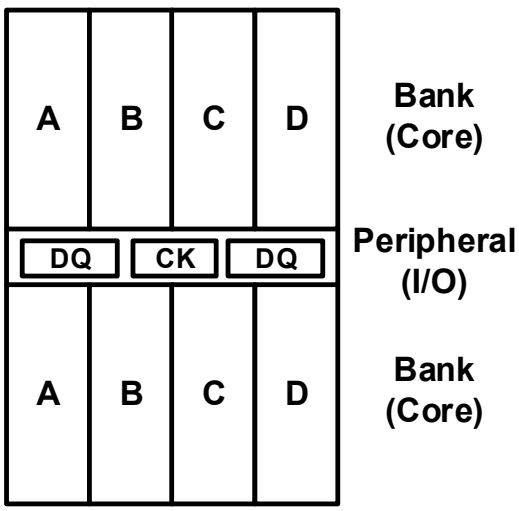


Fig. 2.3 Block diagram of DRAM.

In DQ RX, the data transmitted from the MCU is converted to full swing level. The data is amplified in the full swing in the sampler and aligned by the clock. Figure 2.5 shows a timing diagram of a sampler in the DQ RX. If the clock is not in the center of the data, the timing margin is reduced. Therefore, the write training is implemented to place the clock at the center of the data.

The delay of the input clock path td1 and the output driver td2 varies as process, voltage, and temperature (PVT) change [13]. Thus, the sum of the delay is changed by PVT variation and the data valid window (tDV) decreases as shown in Figure

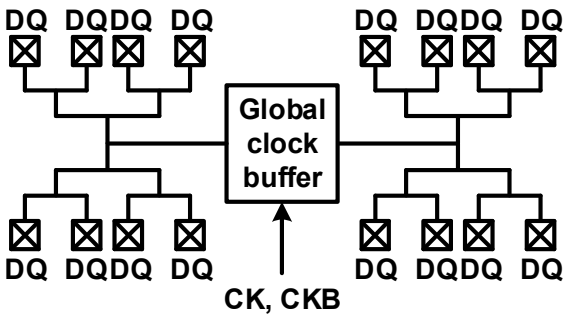


Fig. 2.4 Clock distribution in DRAM.

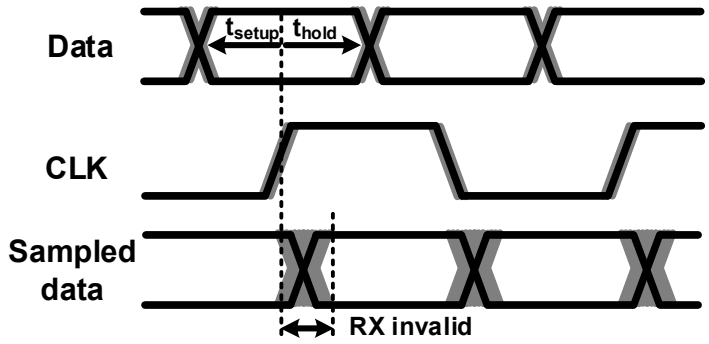


Fig. 2.5 Timing diagram of a sampler in DQ RX.

2.6(b). Thus, the DLL is introduced in the clock path to compensate for PVT variation as shown in Figure 2.6(a). Using a DLL prevents tDV from decreasing since the overall path delay is fixed at N·tCK as shown in Figure 2.6(c).

The DQ TX transmits the data from the DRAM to the MCU. The block diagram and the timing diagram of drivers in the DQ TX are shown in Figure 2.7. In the dif-

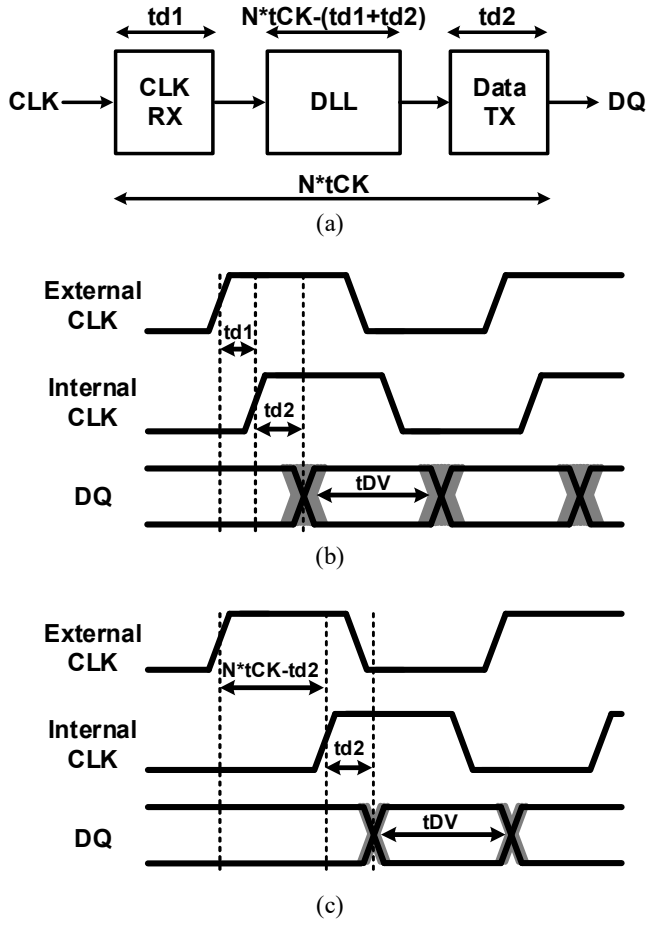


Fig. 2.6 (a) Block diagram of clock path and the timing diagram (b) without DLL and (c) with DLL.

ferential clocking, the data is aligned by differential clocks as shown in Figure 2.7(a). In this architecture, the duty cycle error between the differential clock reduces eye width of the transmitted data. Also, in the quadrature clocking as shown in Figure 2.7(b), the data is aligned by quadrature clocks. Similar to the differential clocking, the phase skew among quadrature clocks degrades eye width. Therefore, it is important to remove the phase skew, so a phase skew corrector is required.

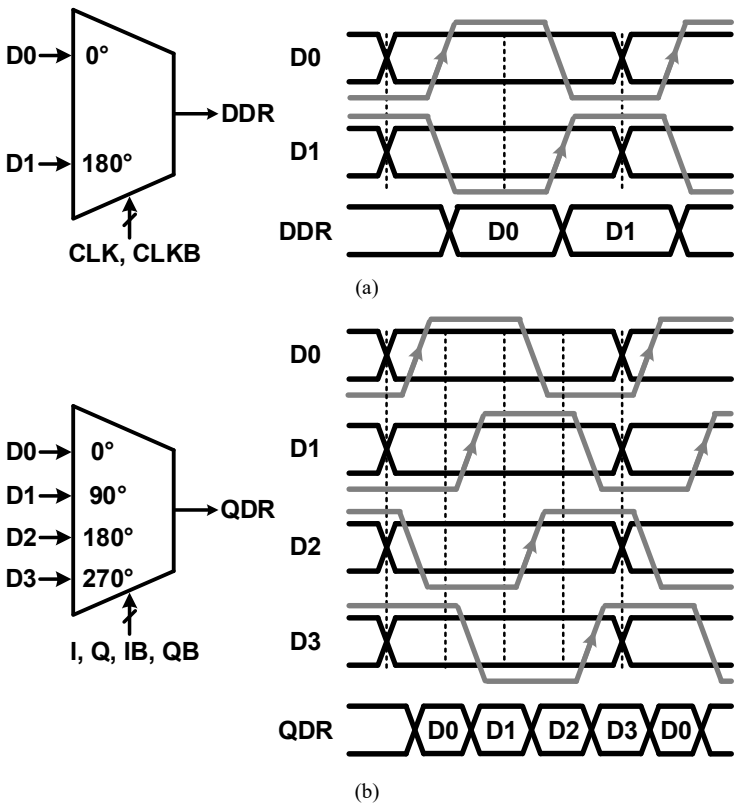


Fig. 2.7 Block diagram and timing diagram of driver in DQ TX (a) differential clocking and (b) quadrature clocking.

Chapter 3

Background on DLL

3.1 Overview

Phase-locked loops (PLLs) are used to generate a high-frequency clock, reduce jitter, and recover the clock in a receiver. Since the PLL contains an oscillator as a clock generator, jitter accumulates. In addition, stability should be considered when designing the system with the PLL due to the second-order system [14]. Like PLL, delay-locked loops (DLLs) are widely used in a clocking of the high-speed interfaces. DLLs are used as a zero-delay buffer in a source-synchronous system, a multiphase generator, a clock multiplier, and so on. The DLL uses delay cell instead of the oscillator. Therefore, it is free of jitter accumulation and stability issue, because it is the one-pole system.

As the PLL has two types, analog PLL and digital PLL, the DLL has two types.

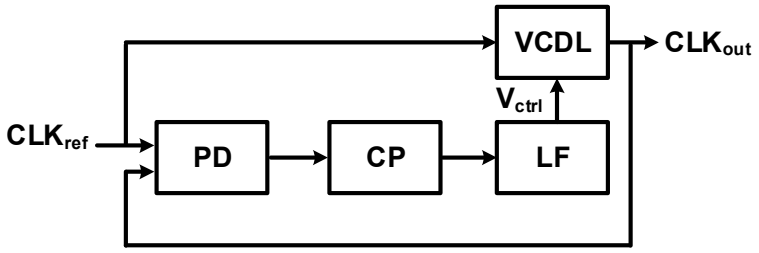


Fig. 3.1 Block diagram of analog DLL.

Figure 3.1 shows the block diagram of an analog DLL. The analog DLL includes a phase detector (PD), a charge pump (CP), a loop filter (LF), and a voltage-controlled delay line (VCDL). The PD compares the phase between the input clock and the feedback clock and outputs a pulse whose pulse width is proportional to the phase difference. Afterward, the CP charges or discharges a capacitor in the LF according to the PD output. The output voltage V_{ctrl} controls the delay of the VCDL. The delayed clock CLK_{out} is fed back to the input of the PD. As a result, the DLL is a negative feedback system where the delayed clock is locked to the reference clock CLK_{ref}. Moreover, the frequency of the output clock and the input clock is the same.

However, the current mismatch in the CP induces error. In addition, due to the process scaling of CMOS technology, the gate leakage current increases and the supply voltage decreases, which makes the design of the analog DLL challenging. Therefore, the all-digital DLL (ADDLL) is introduced as shown in Figure 3.2. ADDLL consists of a PD, a digital loop filter (DLF), and a digitally-controlled delay line (DCDL) which replaces the VCDL of the analog DLL. Contrary to the PD in the analog DLL, the output of the PD in the ADDLL is a digital value that indicates which comes first of the reference clock and the feedback clock. The DLF accumulates and filters the digital output of the PD. The DCDL is controlled by the DLF to

adjust the delay. The DLF occupies a small area and has PVT robustness and better process portability compared to a capacitor, the LF of the analog DLL. However, the ADDLL has a large jitter due to the quantization. The VCDL has fine resolution because it is controlled by a continuous value analog voltage. However, the DCDL is controlled by discrete digital codes. Thus, there are several DCDL architectures to implement a finer resolution.

There are also two types of DLL that are classified according to the presence of another reference source, type-I and type-II. In the type-I DLL, the reference clock and the feedback clock, which is the delayed version of the reference clock, are compared as shown in Figure 3.3. This structure is used for the zero-delay buffer,

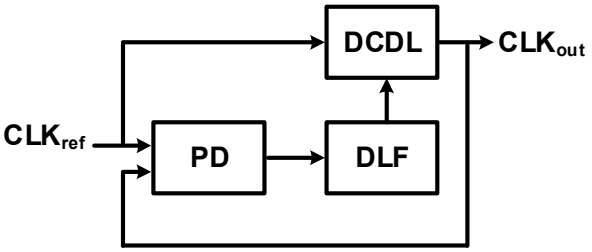


Fig. 3.2 Block diagram of all-digital DLL.

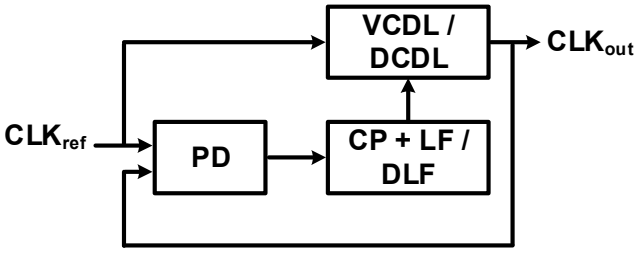


Fig. 3.3 Block diagram of type-I DLL.

multi-phase clock generation, and the clock multiplier.

Figure 3.4 shows the block diagram of the type-II DLL. In the type-II DLL, the reference clock CLK_{ref1} is compared with the feedback clock CLK_{out}, which is the delayed version of the other reference clock CLK_{ref2}. This architecture is utilized for clock generation in source-synchronous systems.

The following section will explain the building blocks of DLLs.

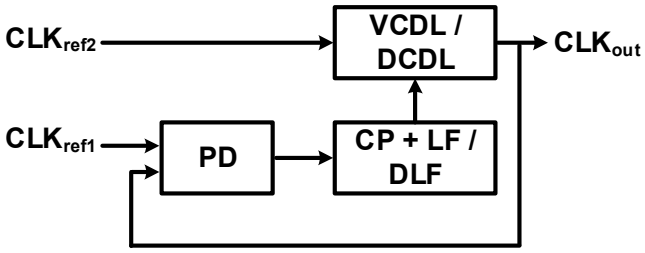


Fig. 3.4 Block diagram of type-II DLL.

3.2 Building Blocks

3.2.1 Delay Line

The delay line has two types, analog and digital. The analog type delay line is controlled by voltage and the digital type delay line is controlled by digital code. The current starved inverter is categorized as VCDL because the voltage controls the current of the inverter, which results in changing the delay, as shown in Figure 3.5. However, the delay curve of the current starved inverter is not perfectly linear, so the gain of VCDL is changes with V_{ctrl}. Also, when limited to the linear range, the available delay range is reduced.

Figure 3.6 shows the shunt capacitor based DCDL. The delay is controlled by adjusting the capacitor with the 3-bit digital code. When the NMOS switch is on, the load capacitor at the inverter output increases and the delay increases. On the other hand, when the NMOS switch is off, the load capacitor does not increase. The delay

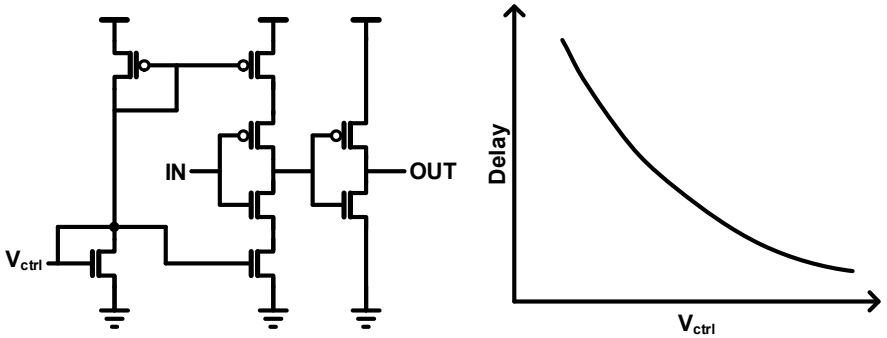


Fig. 3.5 Circuit and delay curve of current starved inverter.

curve of the shunt capacitor based DCDL is linear, but the available delay is quantized.

The NAND gate can be used for the delay line as shown in Figure 3.7. D is a dummy NAND gate to match the load capacitance at the output of each NAND gate. The resolution of the delay line is twice that of the NAND delay ($2 \cdot t_{NAND}$). However, if the control bit changes, there is a glitch problem. In order to address this issue, there are several studies [15], [16].

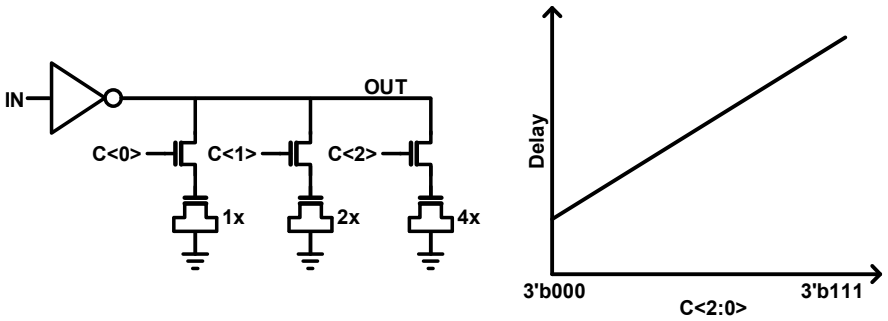


Fig. 3.6 Circuit and delay curve of shunt capacitor based DCDL.

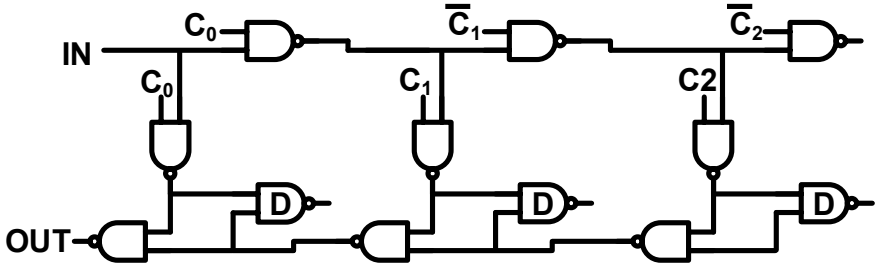


Fig. 3.7 NAND based DCDL.

3.2.2 Phase Detector

Figure 3.8 shows the XOR PD and its transfer curve. The XOR gate is used as a PD and converts the phase difference to the voltage V_{out}. If both inputs (V_1 and V_2) are the same, the output will be zero. Otherwise, the output is non-zero. As a result, the output is minimum when the phase difference is an even multiple of pi and maximum when the phase difference is an odd number multiple of pi. However, if the duty cycle error exists in the input clocks, the transfer curve will be shifted since XOR PD is level-sensitive.

A Bang-Bang PD (BBPD), which is also known as a binary PD, is shown in Fig. 3.9. If the feedback clock V_2 leads the reference clock V_1, the output is -1. Otherwise, the output is +1. It only indicates the sign of phase error without the amount of phase error information. Therefore, its transfer curve is nonlinear and it has infinite gain when the phase error is zero. Also, since the D-flip-flop is edge-sensitive, the duty cycle error does not affect the transfer curve. However, the device mismatch makes offset and shifts the transfer curve.

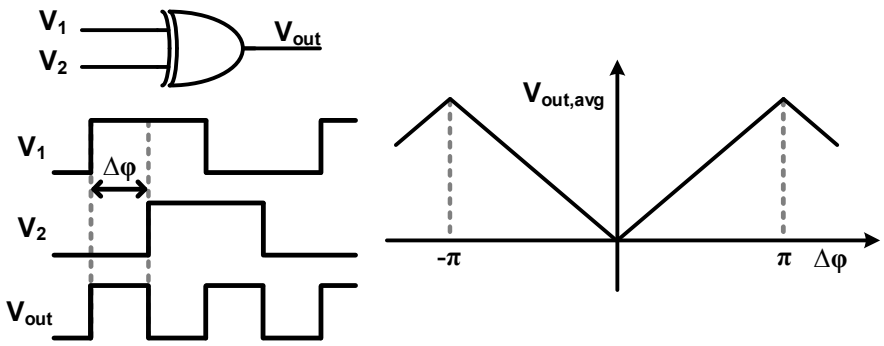


Fig. 3.8 XOR PD and transfer curve.

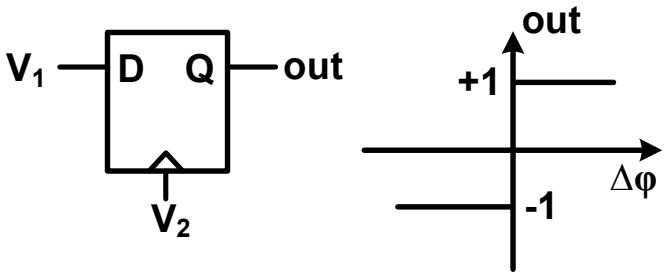


Fig. 3.9 BBPD and transfer curve.

3.2.3 Charge Pump

The CP is used in the analog DLL with the analog PD. Figure 3.10(a) shows the CP circuit. When UP is 1, the charging current flows and V_{ctrl} on a capacitor increases. On the other hand, if DN is 1, the stored charge in the capacitor is discharged and V_{ctrl} decreases. Therefore, the charge pump converts the phase difference information into an analog voltage.

Matching of charge current and discharge current is important because static phase error occurs in the locking state due to the mismatch between currents. As shown in Figure 3.10(b), the charging current and discharging current differ as the control voltage changes. Therefore, single and dual compensation methods are proposed to eliminate the current mismatch [17], [18].

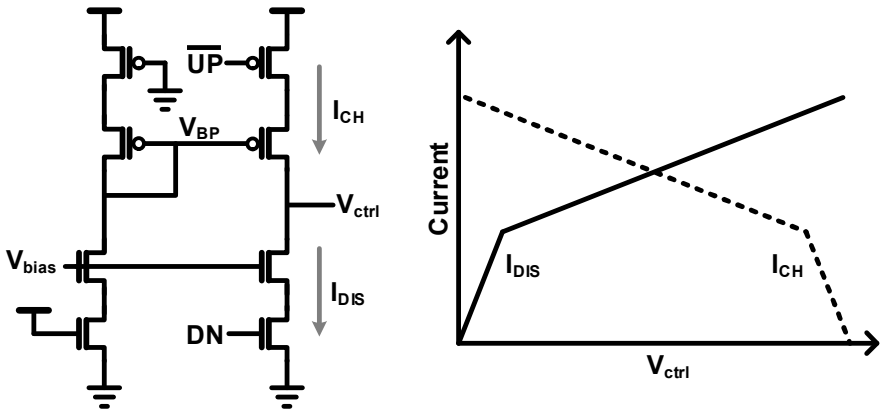


Fig. 3.10 (a) Charge pump circuit and (b) current mismatch.

3.2.4 Loop filter

Figure 3.11 shows the analog and the digital loop filter in the DLL. Unlike the analog PLL, the analog DLL only uses a capacitor as a loop filter. A capacitor stores charge and outputs voltages to control the VCDL. However, the leakage current changes the stored charge and affects the control voltage. Also, the capacitor occupies a large area and has less controllability.

A DLF filters the output of the binary PD and generates the digital control code for the DCDL to remove the phase error. The DLF in the digital DLL accumulates the PD output with programmable gain α. Its function is the same as a capacitor in the analog loop filter. Because the DLF is implemented in a programming language, it has scalability. It also provides better control over the analog loop filter because the gain can be easily adjusted instead of changing the value of the capacitor.

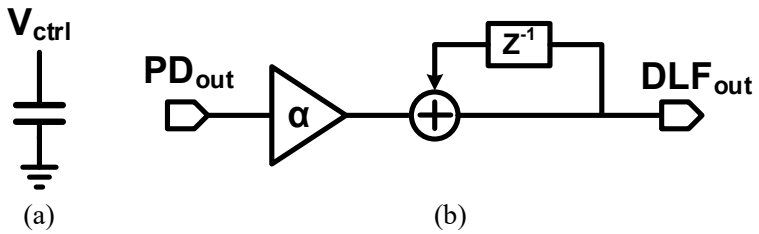


Fig. 3.11 (a) Analog loop filter and (b) digital loop filter.

Chapter 4

Forwarded-Clock Receiver with DLL-based Self-tracking Loop for Unmatched Memory Interfaces

4.1 Overview

The high-speed memory interfaces traditionally use source-synchronous architecture in which a transmitter sends data along with a clock and a receiver latches data with the transmitted clock [19]. Figure 4.1 shows two types of receivers in DRAM. As shown in Figure 4.1(a), the matched type receiver has a tDQS replica delay cell in the data (DQ) path to match the delay of the data strobe (DQS) path. Since the amount of delay variation caused by a voltage or temperature (VT) drift is the same

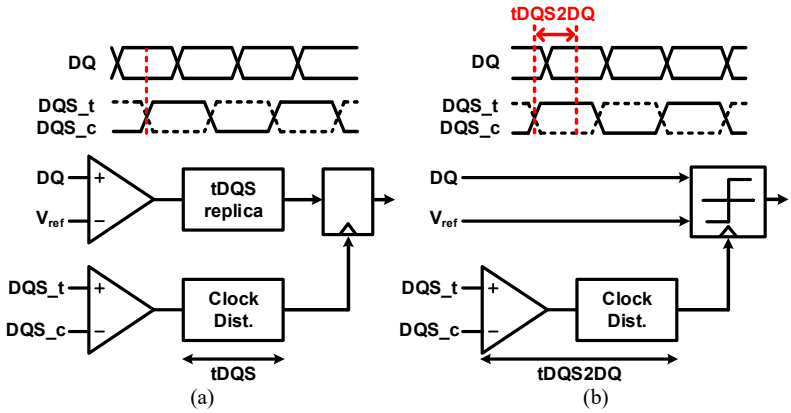


Fig. 4.1 Receiver types: (a) matched and (b) unmatched.

in each path, VT drift does not degrade the timing margin in the matched type receiver. However, recently, the receiver architecture of memory interfaces is moving from the matched type to the 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 Figure 4.1(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 unmatched type receiver, performs a write training to locate a DQS transition edge on the DQ center at the DQ sampler for the optimal timing margin [20]. After the write training is done, a memory controller transmits DQ later than DQS by tDQS2DQ, which means that the relative delay difference between the DQ path and the DQS path. However, VT drift occurring after the write training changes tDQS2DQ and results in the reduced timing margin because the sampling point deviates from the optimal sampling point, the DQ eye center. The reduced timing margin is a critical problem since the one-unit interval (UI) time de-

creases as the data rate increases. Therefore, compensation for the delay drift due to VT variation is required to maintain the timing margin in the receiver.

To detect tDQS2DQ drift, periodic incremental training [3] and an internal DQS clock tree oscillator [2] 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 path delay. From the counted number of errors, the drift of the edge of the DQ eye can be tracked. [2] introduces an internal DQS clock tree oscillator which replicates tDQS2DQ for tracing the drift of tDQS2DQ. To measure the amount of tDQS2Q drift, a counter value synchronized with the oscillator clock is stored in a register for a given time interval in the write training. Then, the entire process is repeated after the write training to observe tDQS2DQ drift by comparing the stored counter values. If a large drift is detected by using the previously mentioned methods, the memory controller is required to perform 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 re-training, embedding a delay-locked loop (DLL) in the DQS path can be a feasible solution for self-tracking because the DLL can fix tDQS2DQ by controlling delay lines of the DQS path even if VT drift occurs. Moreover, the initial write training procedure can be utilized with a DLL since the relative delay between DQ and DQS is set to tDQS2DQ in the initial training. As a result, the DQS transition edge is always centered at the DQ eye and the timing margin is not decreased by VT drift.

Chapter 4. Forwarded-Clock Receiver with DLL-based Self-tracking Loop for Unmatched Memory Interfaces

In this chapter, a power- and area-efficient forwarded-clock (FC) receiver with a DLL-based self-tracking loop is presented [4], which exploits the write training and does not require re-training or DQ transitions for tDQS2DQ drift compensation. This chapter details the architecture and the operation of the proposed FC receiver with a DLL-based self-tracking loop. Then, the measurement results of the proposed scheme are described and compared with previous works.

4.2 Proposed Separated DLL

In the system of LPDDR and DDR, DQS only toggles when DQ is transmitted. In the seamless mode, DQS is consecutive, so the phase comparison between the input DQS (DQS_t or DQS_c) and DQ sampling strobe, DQS_I, can be carried out using a phase detector (PD) as shown in Figure 4.2(a). A feedback-loop in the DLL works using the result of the phase comparison. As a result, tDQS2DQ is fixed as N·UI by means of DLL. Since the optimal sampling point is determined in the write training after tDQS2DQ is fixed by DLL and DLL always keeps tDQS2DQ as N·UI by maintaining the delay of the DQS path, VT drift does not change the sampling point. 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 VT drift occurs. However, if the DQS path delay is too long in the non-seamless mode, the phase comparison between the input DQS and DQS_I, cannot be performed because there is no transition edge in the input DQS when DQS_I has a rising transition as presented in Figure 4.2(b). Thus, another signal (e.g. DQS_edge_t) is required for the phase comparison with DQS_I.

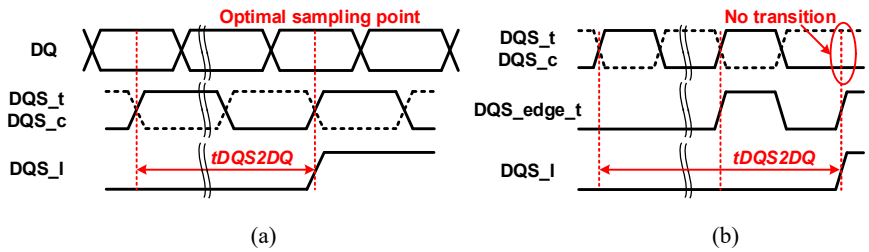


Fig. 4.2 Timing diagram (a) in the seamless mode (b) in the non-seamless mode.

The DLL is separated into two parts to support the burst mode as shown in Figure 4.3. DQS_edge_t and DQS_edge_c are generated in the first-stage DLL for the phase comparison with DQS_I. The first-stage DLL makes DQS_edge signals, which are N_1·UI delayed version of the input DQS, by edge-aligning DQS_edge with the input DQS by controlling digitally-controlled delay lines (DCDL1). The second-stage DLL generates the DQ sampling signal, which is N_2·UI delayed version of DQS_edge, by edge-aligning DQS_I with DQS_edge by adjusting DCDL2. Therefore, the total DQS path delay from the input DQS to DQS_I is equal to N·UI.

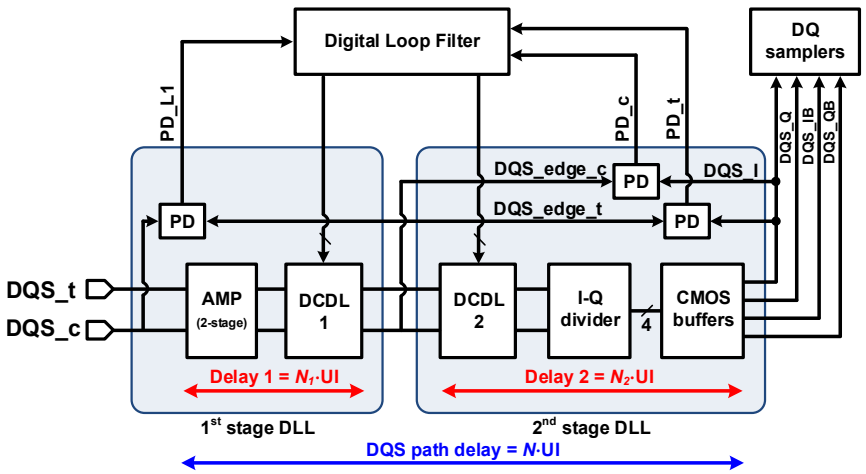


Fig. 4.3 Separated proposed DLL in the DQS path.

4.2.1 Operation of the Proposed Separated DLL

Since DQ sampling signal is produced by delaying the input DQS, the amount of the delay should be minimized for low jitter condition in the locked state. Thus, the first-stage DLL aligns the rising transition of DQS_edge_t with the transition edge of DQS_c closest to the rising transition of DQS_edge_t but is later than the rising transition of DQS_edge_t when the delay of DCDL1 is at the minimum. In the second-stage DLL, the rising transition of DQS_I is aligned with the nearest transition edge of DQS_edge signals that is later than the rising transition of DQS_I when the delay of DCDL2 is minimal.

Figure 4.4 shows the lock point setting method of the first-stage DLL. Under the minimum delay condition of DCDL1, PD_L1 calculates the lock point of the first-stage DLL through phase comparison of DQS_edge_t and DQS_c. The settled 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 delay of DCDL1. Otherwise, if PD_L1 is '1', the rising edge of DQS_edge_t should be aligned with the falling edge

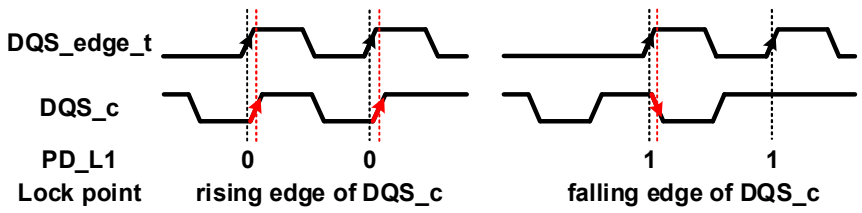


Fig. 4.4 Lock point setting method: 1st stage DLL.

of DQS_c which is the nearest edge from the rising edge of DQS_edge_t. The determined lock point in the minimum delay condition can be varied depending on the initial delay condition or PVT variations due to the different positions of the rising edge of DQS_edge_t. Similarly, the lock point of the second-stage DLL is determined as the rising transition of DQS_edge which is closest to the rising transition of DQS_I when the delay of DCDL2 is at the minimum for the same reason. The rising edge of DQS_I can be aligned with the rising edge of DQS_edge_t or DQS_edge_c by increasing the 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 Figure 4.5, in the case shown on the left, (PD_c, PD_t) = (0, 1) pattern is detected and the rising edge of DQS_I is between the rising edges 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, (PD_t, PD_c) = (0, 1) pattern is detected and the rising edge of DQS_I should be aligned with the rising edge of DQS_edge_c by increasing the delay of DCDL2.

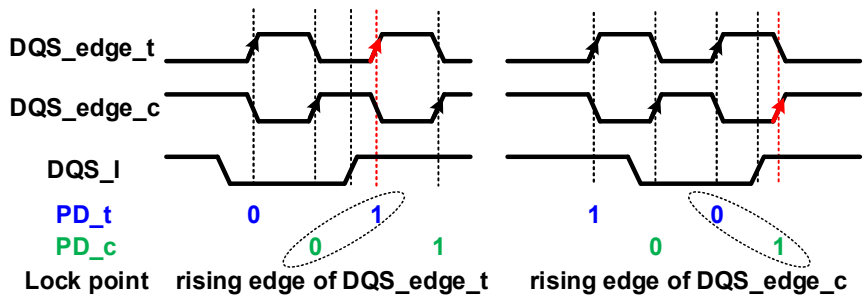


Fig. 4.5 Lock point setting method: 2nd stage DLL.

After the lock point setting, DCDL in each stage DLL is controlled by the digital loop filter (DLF). The rising transition of DQS_edge or DQS_I is can be changed due to VT drift. The determined lock point and PD outputs indicate the current position of the rising edge of signals to be aligned and inform the DLF of the direction to update DCDL. In Figure 4.6, if the lock point of the first-stage DLL is the rising edge of DQS_c and the rising edge of DQS_edge_t is later than the rising edge of DQS_c, the update polarity of the code of DCDL1 is DN. Otherwise, DQS_edge_t leads DQS_c, the update polarity of the code of DCDL1 is UP to align the rising edge of DQS_edge_t to the lock point by increasing the code of DCDL1. However, if the lock point is the falling edge of DQS_c, the update polarity of the code of DCDL1 is reversed even if the rising edge of DQS_edge_t is in the same area.

Also, the code of DCDL2 is controlled by the location of the DQS_I rising transi-

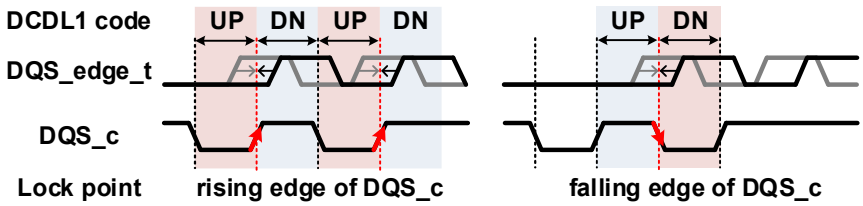


Fig. 4.6 Update polarity of DCDL1 code in the 1st stage DLL.

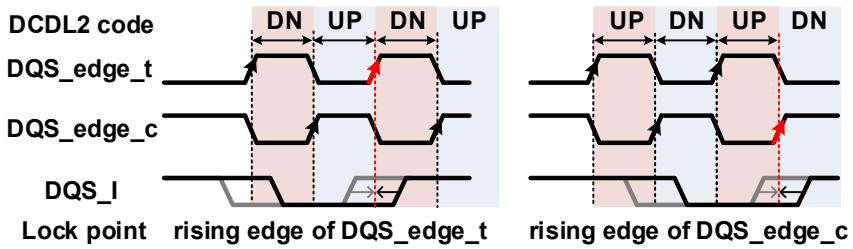


Fig. 4.7 Update polarity of DCDL2 code in the 2nd stage DLL.

tion and the lock point of the second-stage DLL as shown in Figure 4.7. When the lock point of the second-stage DLL is the rising transition of DQS_edge_t and the rising edge of DQS_I is later than the lock point, the update polarity of the code of DCDL2 is DN. Otherwise, the update polarity of the code of DCDL2 is UP to increase the DCDL2 code to align the rising edge of DQS_I with the lock point. As with the first stage DLL, if the lock point is different, the update polarity of the code of DCDL2 is inverted in the second stage DLL.

Consequently, the DLF controls DCDLs to make the DQS path delay from the input DQS to DQS_I as N·UI. If the duty cycle of the input DQS is not 50%, the DQS path delay will not be the same as N·UI. However, only fixing the DQS path delay is important since the fixed sampling point will be at the optimal sampling point when the memory controller performs the write training.

4.2.2 Operation of the Digital Loop Filter in DLL

Figure 4.8 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^2C for adjusting the locking time of each DLL. First, the lock point of each 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 the initial value which is calculated at the lock point setting step. For example, if the PD_L1 is '0' when the coarse delay of DCDL1 is at the minimum, the PD_L1 can

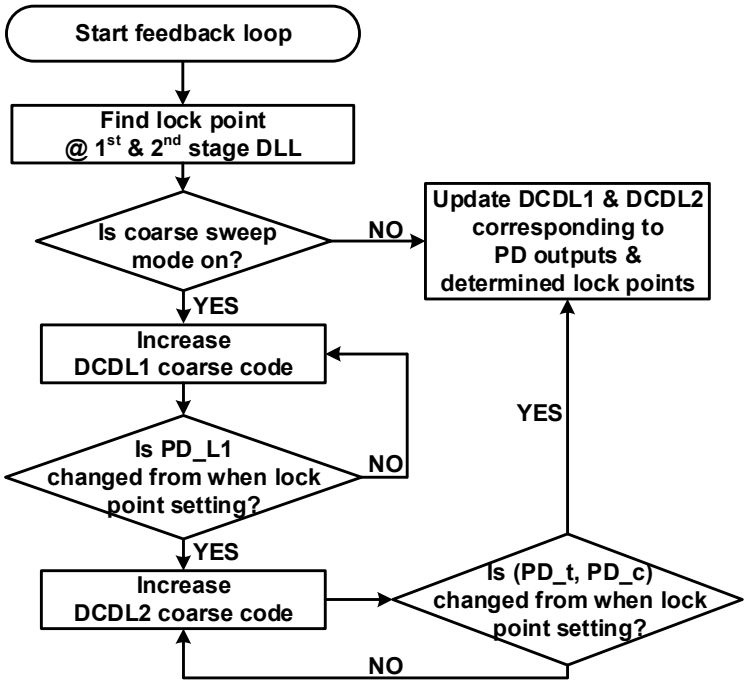


Fig. 4.8 A flow chart of the digital loop filter operation.

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 because the rising edge of DQS_edge_t is later than the lock point, the rising edge of DQS_c. The following step is the coarse sweep of the second-stage DLL. The coarse sweep of the second-stage DLL is done in a similar way with the first-stage DLL. Then, the codes of DCDL1 and DCDL2 are updated simultaneously corresponding to the PD outputs and the determined lock points as mentioned in Section 4.2.1.

4.3 Circuit Implementation

Figure 4.9 shows the overall architecture of the proposed FC receiver. The FC receiver consists of a DLF, 1:4 deserializers (DES), a divider, DQ samplers, and two-stage cascaded DLLs which include PDs and DCDLs. Since the input DQS 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 as shown in Figure 4.10(a). Since the input DQS is the sampler's input and DQS_edge_t is used as the sampling clock, PD_L1 becomes '0' when sampling the zero value of DQS_c. On the other hand, as shown in Figure 4.10(b), a D-flip-flop (DFF) is employed as the PD of the second-stage DLL since both DQS_edge signals and DQS_I are full swing signals.

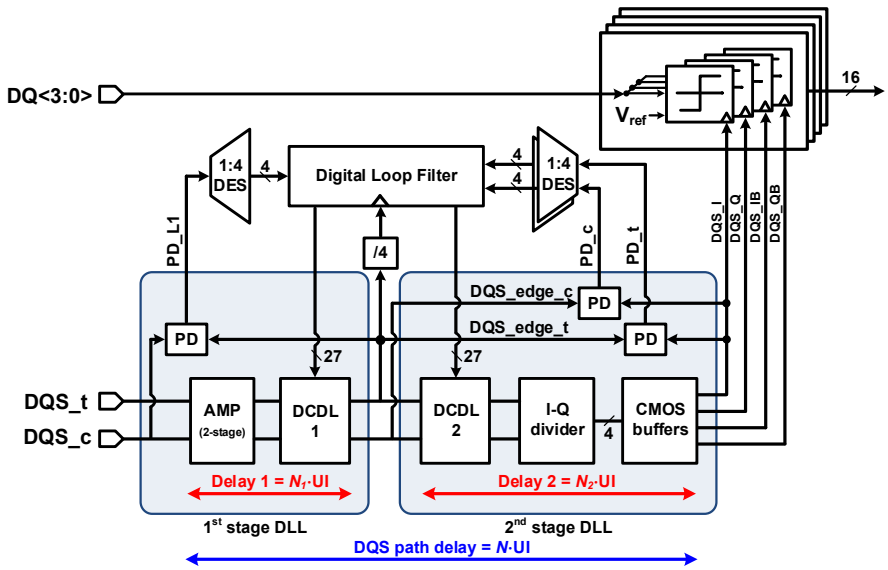


Fig. 4.9 Overall architecture of the proposed forwarded-clock receiver.

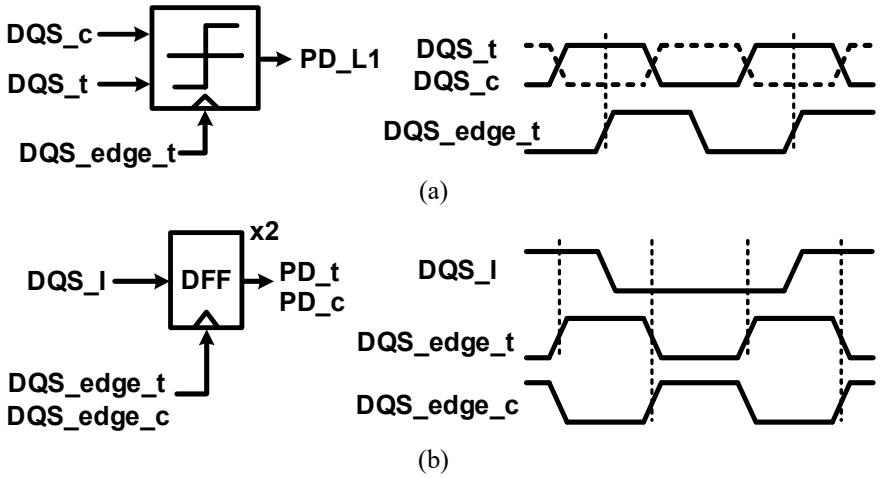


Fig. 4.10 Architecture and the timing diagram of the phase detector (a) in the 1st stage DLL (b) in the 2nd stage DLL.

The output of the PD is PD_t when the sampling clock is DQS_edge_t and PD_c when the sampling clock is DQS_edge_c. The DQS path includes a two-stage amplifier, two DCDLs, an I-Q divider, and CMOS buffers. The two-stage amplifier amplifies the input DQS from the 0.4-V peak-to-peak swing to the rail-to-rail swing.

As shown in Figure 4.11, DCDL is composed of dual coarse delay lines and a phase interpolator (PI) based on tri-state inverters [13] to cover a wide delay range larger than 1 UI with fine resolution while avoiding a boundary switching problem. In the coarse delay line, 6-bit thermometer codes CU and CD control the number of on-state NANDs in CLKU and CLKD paths, respectively, which exhibit a $2 \cdot t_{NAND}$ time difference between CLKU and CLKD. The PI interpolates CLKU and CLKD by adjusting 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 DCDL has an effective resolution, $2 \cdot t_{NAND}/15$.

Figure 4.12 shows the seamless boundary switching in DCDL. The boundary switching problem that introduces large jitter occurs when the codes of both coarse and fine delay lines are updated at the same time. To remove the issue, when the delay of DCDL in the proposed DLL is increased in the boundary case, FSEL is not changed but the code of the coarse delay line, CU (or CD) is updated with maintaining the same delay in the output. As a result, the delay from IN to OUT is increased seamlessly.

The I-Q divider generates 4-phase clocks for a quarter-rate clocking. CMOS

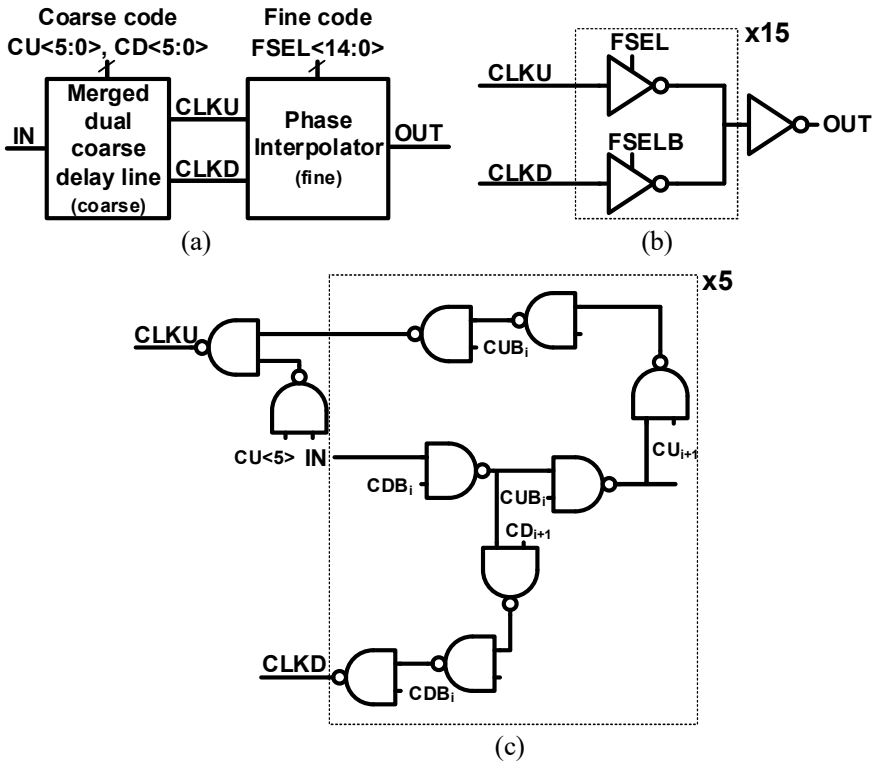


Fig. 4.11 (a) DCDL architecture (b) phase interpolator (c) merged dual coarse delay line.

buffers distribute quadrature clocks to four samplers per each phase. The DLF adjusts both DCDL1 and DCDL2 to fix the total DQS path delay as N times UI by two-stage cascaded DLLs.

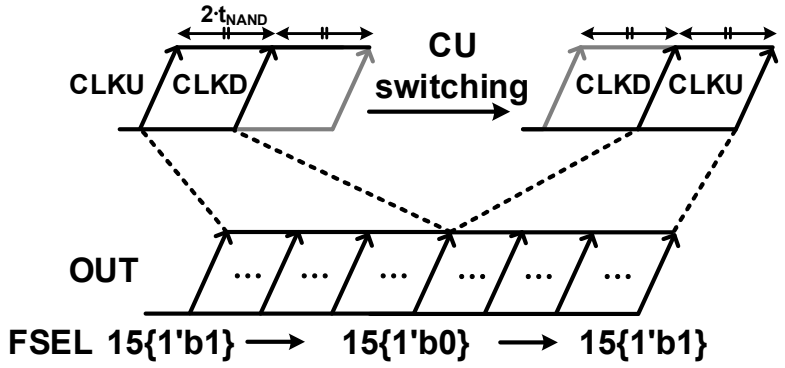


Fig. 4.12 Seamless boundary switching in DCDL of the proposed DLL.

4.4 Measurement Results

The proposed FC receiver (RX) is fabricated in the 65-nm CMOS technology. The chip microphotograph and the core layout of the FC RX are shown in Figure 4.13. 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 [21]. Using standard cells, the DLF is fully synthesized and automatically placed and routed. The active area of the proposed FC RX is 0.0329 mm^2, which includes 4 DQ lanes.

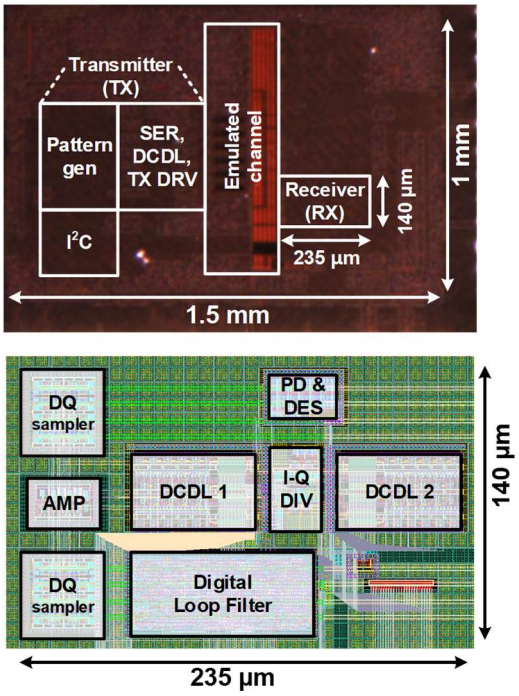


Fig. 4.13 Chip microphotograph and core layout of the proposed scheme.

4.4.1 Measurement Setup and Sequence

Figure 4.14 shows the measurement setup of the proposed FC RX. 6.4 GHz clock from J-BERT is transmitted to the pattern generator of the TX for the generation of 3.2 GHz the input DQS and 6.4 Gbps PRBS7 DQ. The sampler's output data in the RX is connected to the input of the error detector. PC controls I^2C and the supply voltage of the RX. Also, PC receives the bit error rate (BER) from the error detector while sweeping the DCDL codes with I^2C.

Figure 4.15 shows the measurement sequence of the proposed scheme [21]. First, set VDD as 1 V and start the DLL feedback loop to fix the total DQS path delay as N·UI. Next, fix the codes of the TX DCDL in the DQ path to make the sampling point is the optimal. Then, change VDD and hold the RX domain DCDL codes after

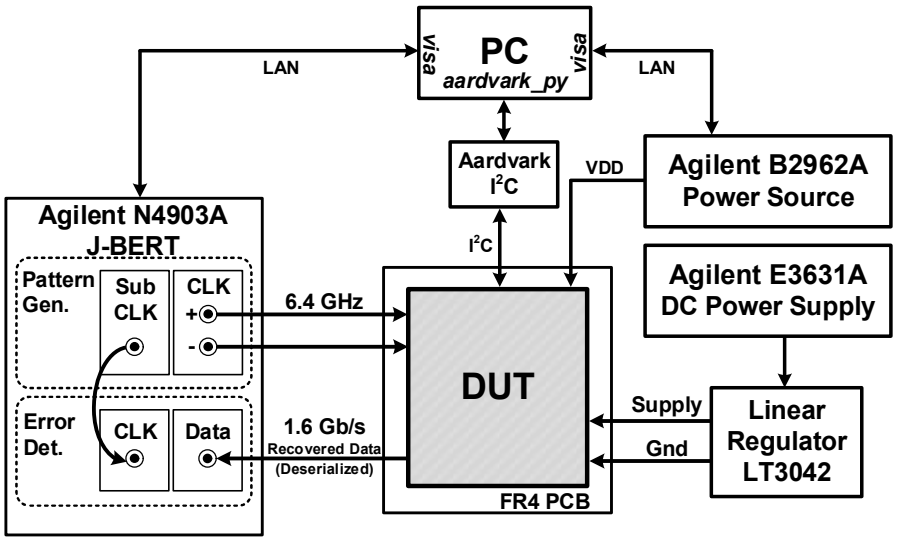


Fig. 4.14 Measurement setup.

the loop is settled. Next, measure BER while adjusting the TX DQS delay by sweeping the TX DCDL of the TX DQS path to change the sampling point.

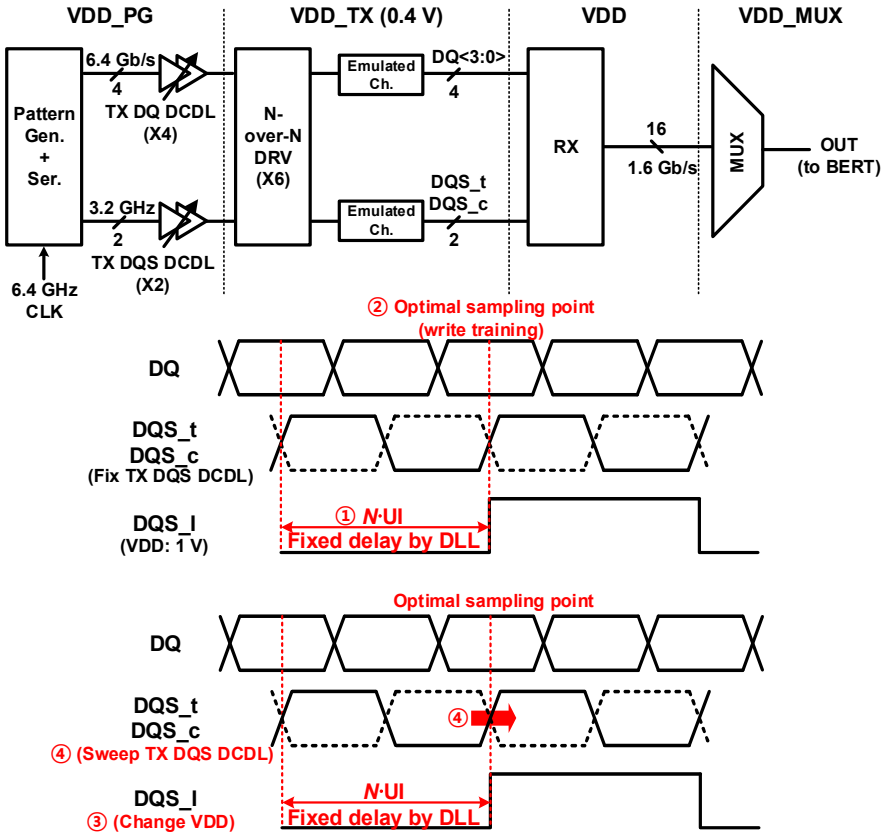


Fig. 4.15 Measurement sequence.

4.4.2 VT Drift Measurement and Simulation

The graph of timing margin versus supply voltage VDD is shown in Figure 4.16. The DLL-off case is explored when all DCDL codes of RX domain 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 in 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.

Figure 4.17 and Figure 4.18 shows the measured DQ eye diagram and the bath-tub curve at three different supply voltages when the write training is carried out at 1 V. Contrary to the DLL-off case, in the DLL-on case, both the DQ eye and the sampling point do not shift because the sampling point is not modified.

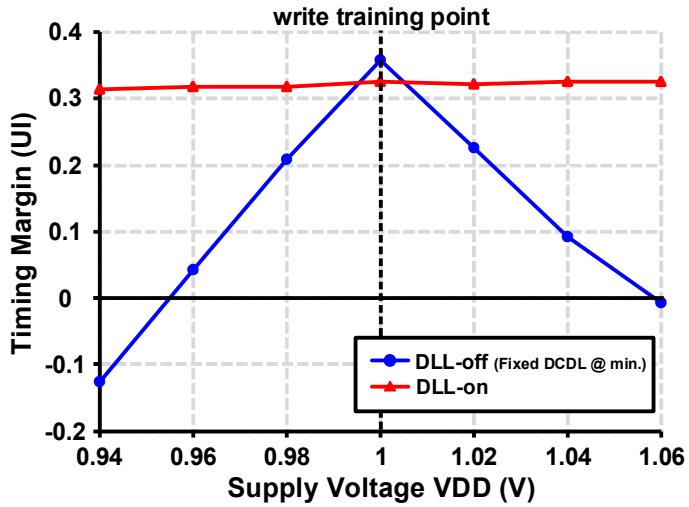


Fig. 4.16 Measured timing margin vs. VDD variation (PRBS7, BER=10^{-9}).

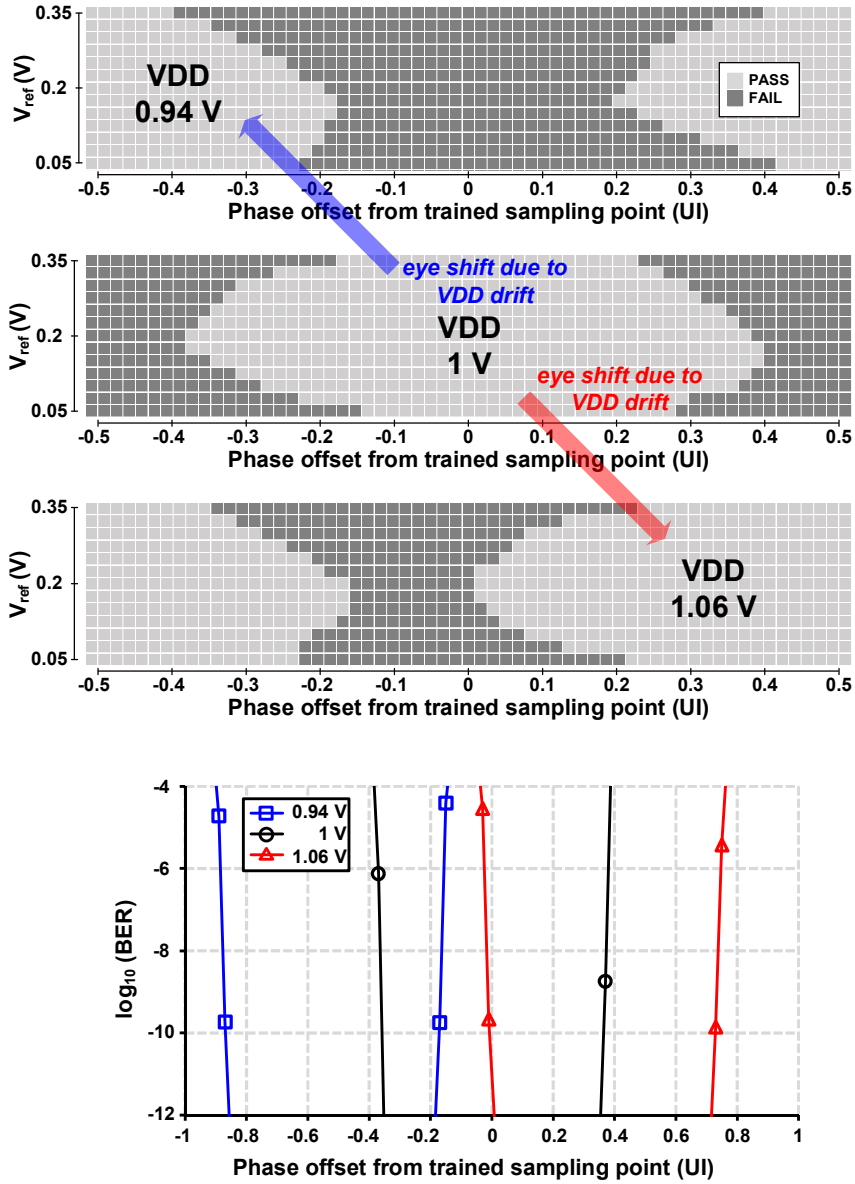


Fig. 4.17 Measured DQ eye diagram (PRBS7, BER=10^{-9}) and bathtub curve at 0.2-V V_{ref} of DLL-off case.

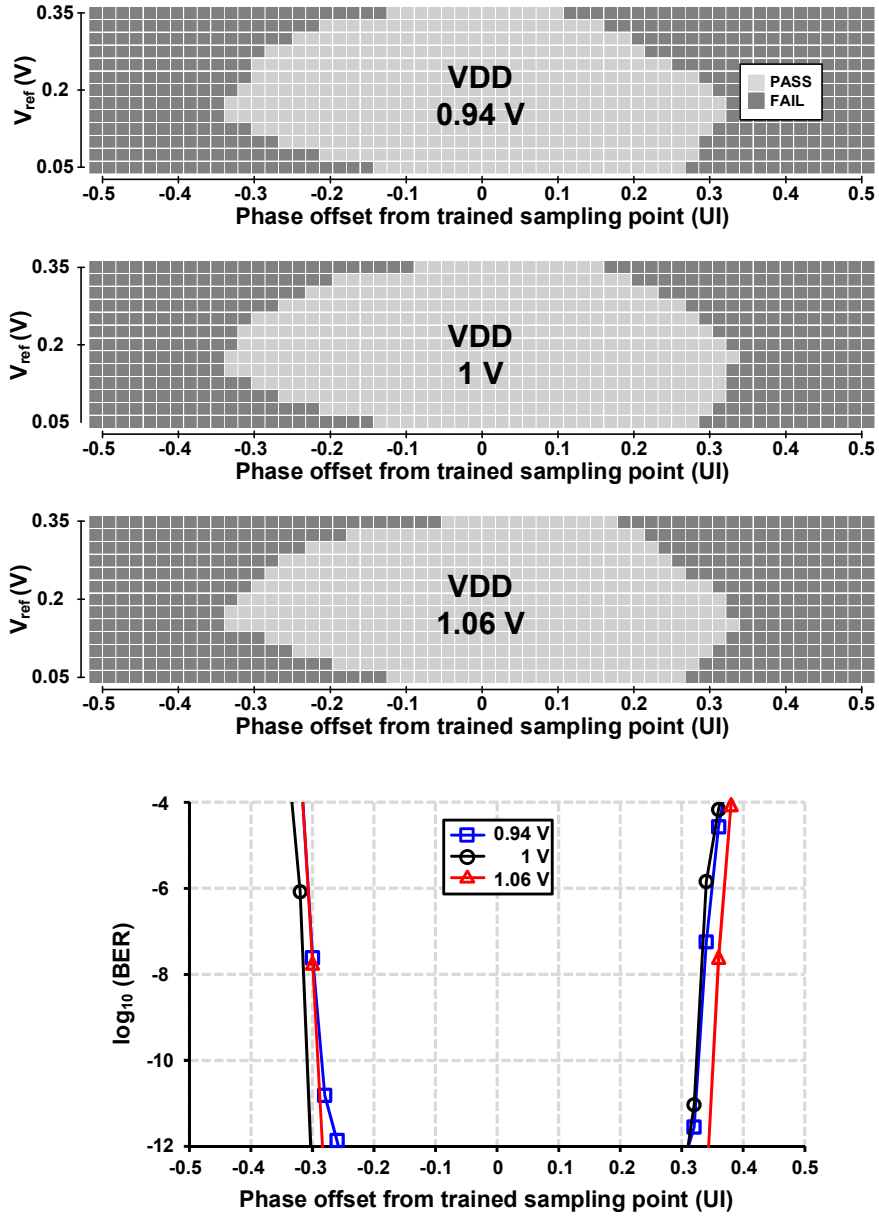


Fig. 4.18 Measured DQ eye diagram (PRBS7, BER=10^{-9}) and bathtub curve at 0.2-V V_{ref} of DLL-on case.

The simulated tracking behavior of the proposed scheme when the 0.06 V supply voltage drift or 60 °C temperature drift occurs from 200 ns to 300 ns is shown in Figure 4.19. When the DCDL update gain is high, the sampling time difference from the settled point remains constant due to a high loop bandwidth.

Figure 4.20 shows the measured power breakdown of the proposed FC RX. The

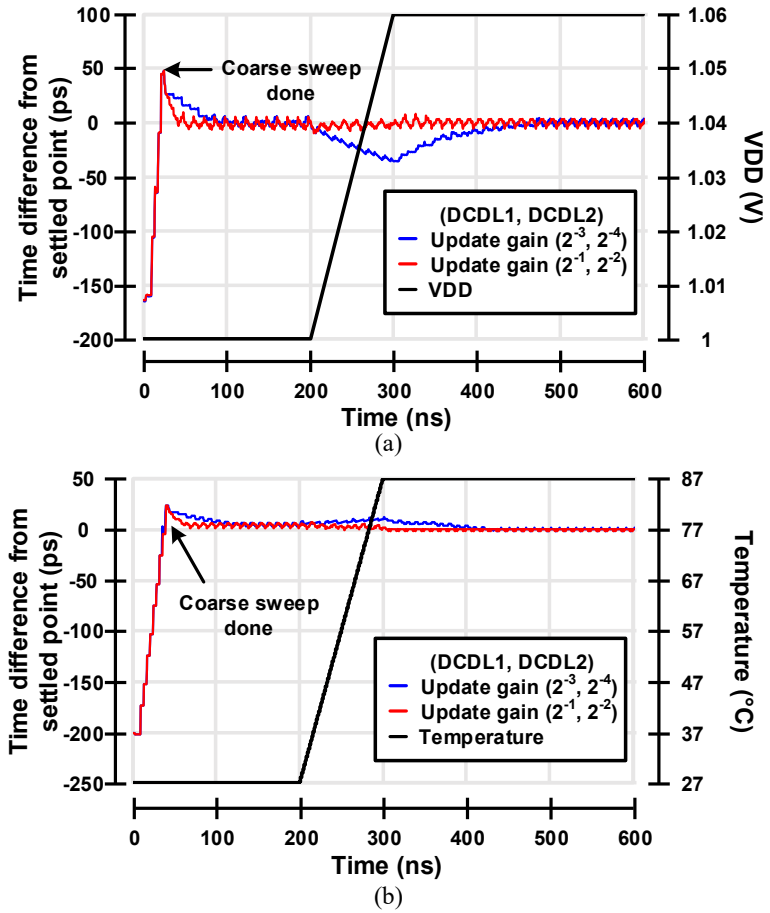


Fig. 4.19 Simulated tracking behavior of the proposed FC receiver (a) when voltage drift and (b) when temperature drift occur.

total power consumption is 11.45 mW from 1-V supply at 6.4 Gb/s including 4 DQ lanes. In Table 4.1, performance of the proposed FC receiver is summarized and compared with those of the state-of-the-art FC receivers [21]-[26]. The proposed FC receiver occupies a small area and consumes 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 fixed sampling point as the optimal sampling point for the DQ eye centering.

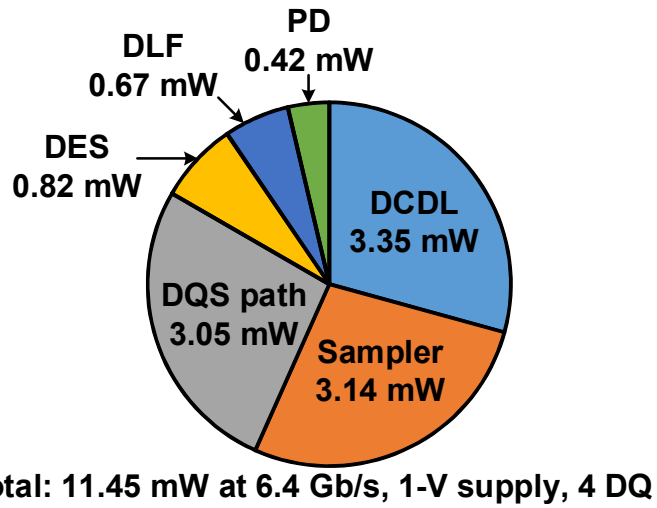


Fig. 4.20 Measured power breakdown of the proposed receiver.

Table 4.1 Performance comparison

	ISSCC'12 [22]	CICC'13 [23]	SOVC'14 [24]	TCAS2'16 [25]	TCAS1'16 [26]	SOVC'19 [21]	This work
Technology	90 nm	28 nm	65 nm	65 nm	65 nm	65 nm	65 nm
Architecture	Digital DLL	Digital DLL	ILO + 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*	6.73*	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 (mm^2)****	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
**With equalizer power
***Includes 4 data lanes
****Area per lane
*****Includes TX + RX

Chapter 5

Open-loop-based Voltage Drift Compensation in Clock Distribution

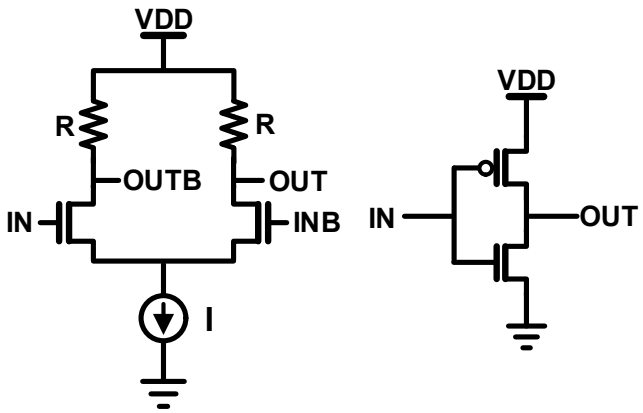
5.1 Overview

In the DRAM interface, the differential clock is in the center of the chip and goes into a transceiver of DQ as shown in Figure 2.4. Therefore, the clock is distributed as an H-tree and runs a distance of a few millimeters using buffers. As the frequency of the clock increases, a current mode logic (CML) buffer is used instead of a CMOS buffer in the step before the differential clock is divided into a lower frequency. After that, the CML level is converted to the CMOS level in the CML-to-CMOS converter (C2C) for use in the transceiver.

Two types of clock buffers are shown in Figure 5.1. The CML buffer draws static

current and a single-ended peak-to-peak (P-P) swing is IR.

Unlike the CML buffer, the CMOS buffer, which is an inverter, consumes current when the inverter input is switched because the MOSFET is turned on only



(a) (b)

Fig. 5.1 Two types of clock buffer: (a) CML and (b) CMOS.

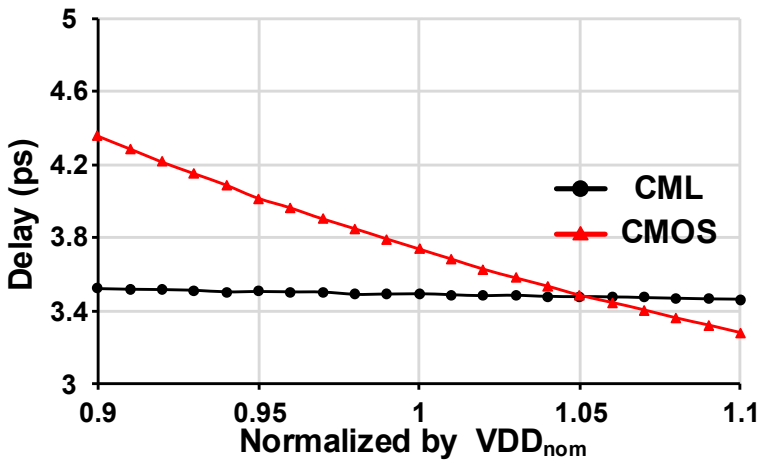


Fig. 5.2 Delay of CML and CMOS when supply voltage drifts.

when the input is in transition. It also consumes static current due to leakage, but the amount of the leakage current is negligible. The P-P swing of the CMOS buffer is rail-to-rail. Figure 5.2 shows the delay of buffers while sweeping the supply voltage VDD. The CMOS buffer is more sensitive to voltage drift than the CML buffer because current and the P-P swing changes as VDD changes. The delay sensitivity of the CMOS buffer is commonly known as 1%-delay/1%-supply [27]. Unlike the inverter, the delay of the CML buffer is $\Delta t_d = C\Delta V_{P\text{-}P}/I = RC$, so it depends only on the load resistance (R) and the load capacitance (C).

Moreover, since the threshold voltage (V_{th}) of the MOSFET changes with temperature as shown in Figure 5.3, the delay also changes with temperature changes. However, the mobility of electrons reduces as temperature increases, so the change in delay due to temperature drift is not dominant than voltage drift as shown in Fig-

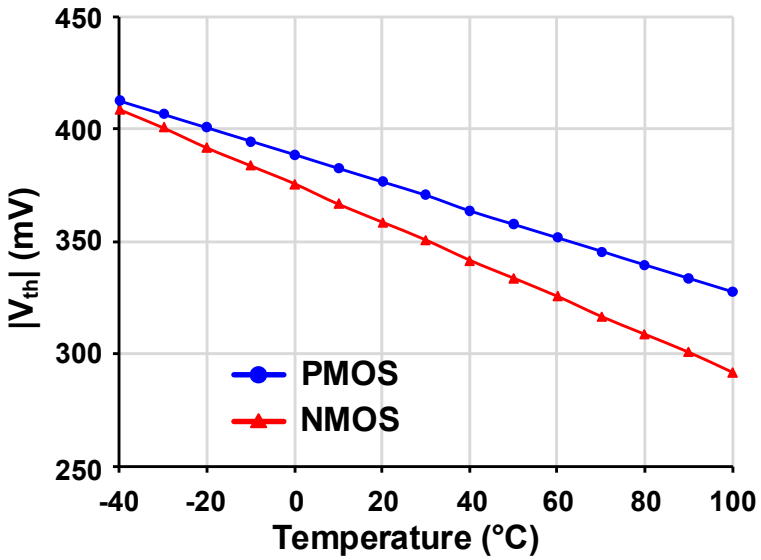


Fig. 5.3 $|V_{th}|$ vs. temperature in PMOS and NMOS.

ure 5.4.

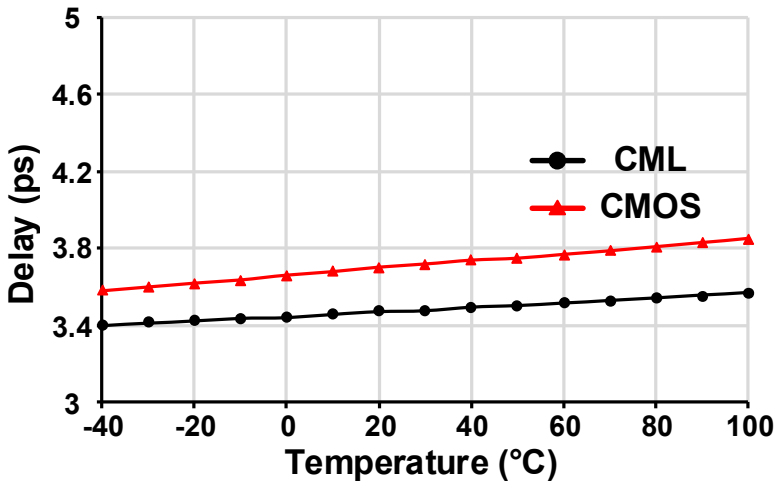


Fig. 5.4 Delay of CML and CMOS when temperature drifts.

5.2 Prior Works

There are several studies on eliminating supply noise or temperature drift. In Figure 5.5, the latch and the resistor are inserted at the output of the ring oscillator [28]. The resistor reduces positive supply sensitivity and the latch has negative supply sensitivity. As a result, the supply sensitivity of the oscillator can be zero at a certain supply voltage, but the method sacrifices the speed of the oscillator.

A drift compensation scheme using a bandgap reference (BGR) is proposed [5]. In Figure 5.6, the control voltage V_{gap} generated by the BGR controls the PMOS resistance to adjust the effective capacitance at the output of the clock buffer for delay drift compensation. V_{gap} is a constant voltage, so as VDD increases, V_{SG} increases. Therefore, the PMOS resistor decreases and the effective load capacitor increases, resulting in positive supply sensitivity of the compensator circuit. Since V_{gap} is cre-

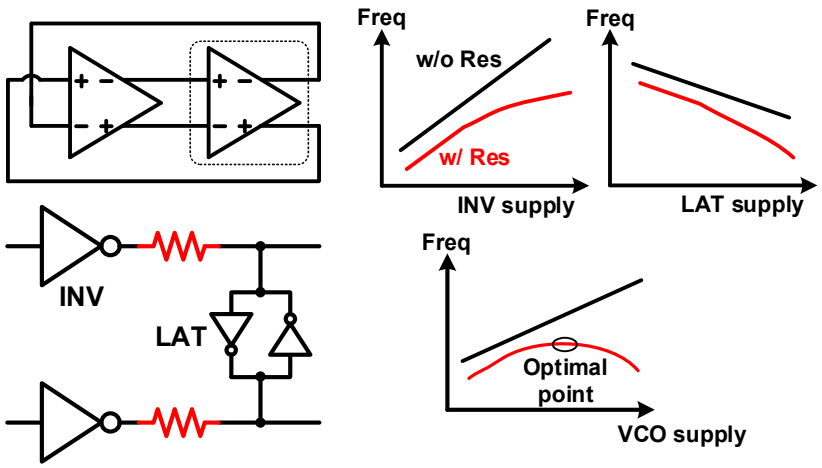


Fig. 5.5 Supply-noise insensitive ring oscillator.

ated to track process and temperature fluctuations, this scheme can compensate for temperature drift and process variation as well.

Figure 5.7 shows a delay circuit that is insensitive to supply and temperature [6]. The RC time of the delay circuit is fixed by designing R1 as the dominant resistor, which means R1 is larger than MOS R. When the temperature rises, P2 turns on early due to V_{th} reduction to compensate for the low saturation current of the inverter.

However, these approaches slow down the clock path due to the increased R or C.

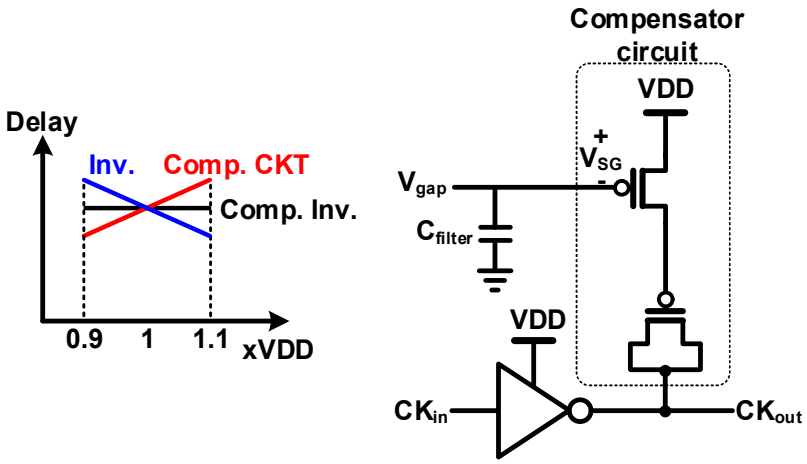


Fig. 5.6 Supply insensitive clock buffer.

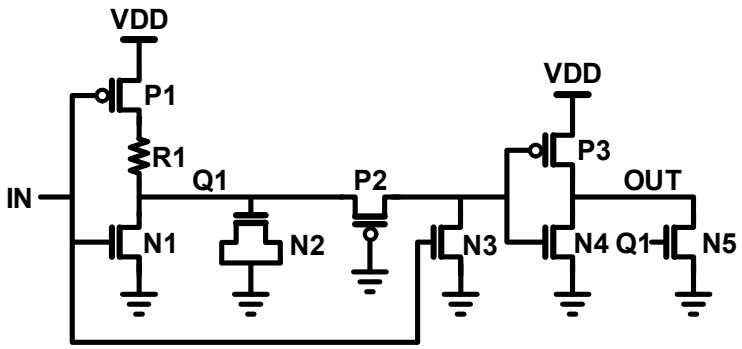


Fig. 5.7 Supply and temperature insensitive delay circuit.

5.3 Voltage Drift Compensation Method

The overall clock distribution tree is shown in Figure 5.8. The global clock is distributed as the CML level and converted to the CMOS level in the local network. A DLL-based scheme is proposed to compensate for delay fluctuations due to VT drift [4]. However, it occupies more area and consumes more power than the open-loop method. In the open-loop scheme, supply noise can be canceled by adjusting the delay in both the CML path and the CMOS path. The CML delay can be adjusted with the load resistance, but changing the load resistance alters the DC operating point, making circuit design difficult. Instead of controlling R, inserting a load capacitor at the CML output degrades the bandwidth. Thus, the CMOS delay circuit handles the overall delay variation in the clock path with overcompensation to cover delay drift in both the CML and the CMOS paths.

There are two types of CMOS delay circuits: VCDL and DCDL. Since the delay is quantized in the DCDL, the VCDL is chosen as a CMOS delay cell to compensate

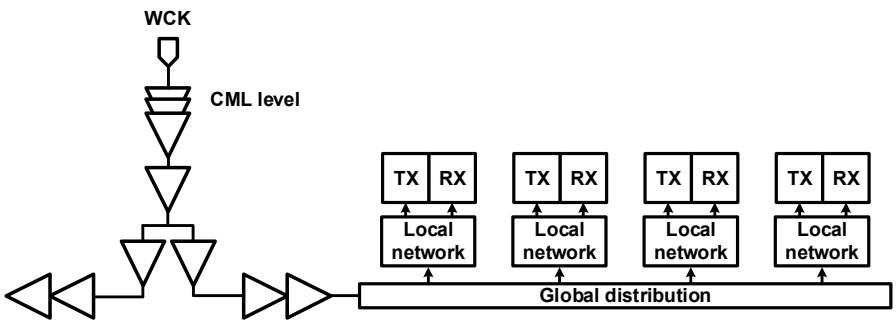


Fig. 5.8 Block diagram of clock distribution tree.

for delay drift in the clock distribution path.

In order to minimize the increased capacitor in terms of power consumption, a current starved inverter is used as shown in Figure 5.9. Biasp controls the pull-up current and biasn controls the pull-down current of the inverter. The PMOS and the NMOS, which are always on, prevent low slope of OUT. To counteract the negative supply sensitivity of the inverter delay, the pull-up and the pull-down current should be reduced when VDD increases. Thus, as the supply voltage increases, $|V_{gs}|$ of both

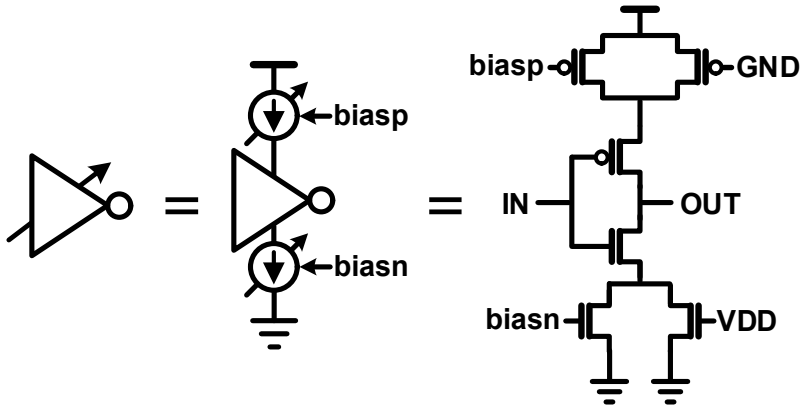


Fig. 5.9 Current starved inverter controlled by bias voltages.

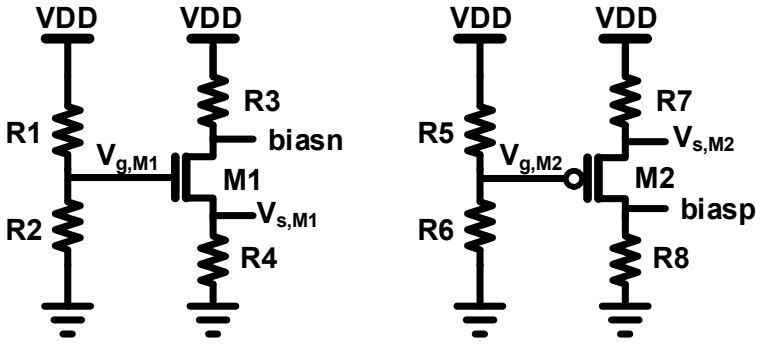


Fig. 5.10 Bias generation circuit for voltage drift compensation.

PMOS and NMOS should be reduced. Figure 5.10 shows the bias generation circuit for voltage drift compensation, which creates biasn and biasp. The supply voltage VDD and the current in M1 I_{M1} affect biasn as expressed in (5.1). By making the increased amount in $I_{M1}R3$ greater than in VDD, biasn decreases as VDD increases. Since I_{M1} depends on $V_{gs,M1}$, $V_{gs,M1}$ should increase as VDD increases.

$$biasn = VDD - I_{M1} \times R3 \tag{5.1}$$

The $V_{g,M1}$ is generated by dividing the supply voltage by poly resistors, so that $V_{g,M1}$ has positive supply sensitivity and is expressed as

$$V_{g,M1} = VDD \times \frac{R2}{R1 + R2} \tag{5.2}$$

Similar to biasn, I_{M2} affects biasp and biasp is expressed as

$$biasp = I_{M2} \times R8 \tag{5.3}$$

As VDD increases, $|V_{gs,M2}|$ should be increased to increase biasp. It is the best to reduce $V_{g,M2}$ when VDD increases, but another MOSFET is required. Therefore, $V_{g,M2}$ is generated by dividing VDD like $V_{g,M1}$ in the generation of biasn. $V_{s,M2}$ and $V_{g,M2}$ are expressed in (5.4) and (5.5). $|V_{gs,M2}|$ is calculated in (5.6).

$$V_{s,M2} = VDD - I_{M2} \times R7 \tag{5.4}$$

$$V_{g,M2} = VDD \times \frac{R6}{R5 + R6} \tag{5.5}$$

$$|V_{gs,M2}| = VDD \times \frac{R5}{R5 + R6} - I_{M2} \times R7 \tag{5.6}$$

The gain of the bias generator can be adjusted by the MOSFET size and resistance. The supply sensitivity of biasn and biasp are shown in Figure 5.11.

The AC-coupling C2C consists of an AC-coupling capacitor, an inverter, and a feedback resistor [29]. Figure 5.12 shows that the inverter and the feedback resistor are controlled for delay adjustment. In general, a metal-oxide-metal (MOM) capaci-

tor is used as a coupling capacitor, but the MOM capacitor is not available in the DRAM process due to the small number of available metals. Therefore, the NMOS capacitor is used for the proposed C2C. The current starved inverter and the MOS resistor (MOS R) are used to adjust the delay according to VDD fluctuations. In the MOS R, the PMOS transistor and the NMOS transistor are connected in parallel and the gate voltages are controlled to adjust the resistance of MOS R. Considering the supply sensitivity of biasn and biasp, MOS R increases when VDD increases. Since

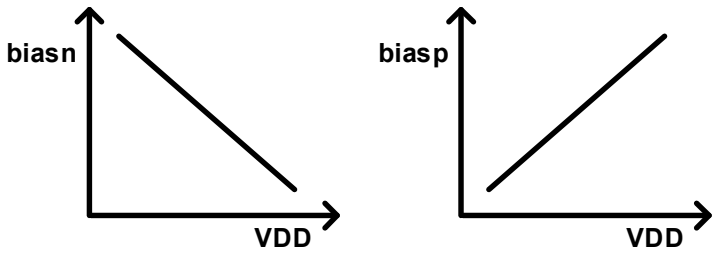


Fig. 5.11 Supply sensitivity of biasn and biasp.

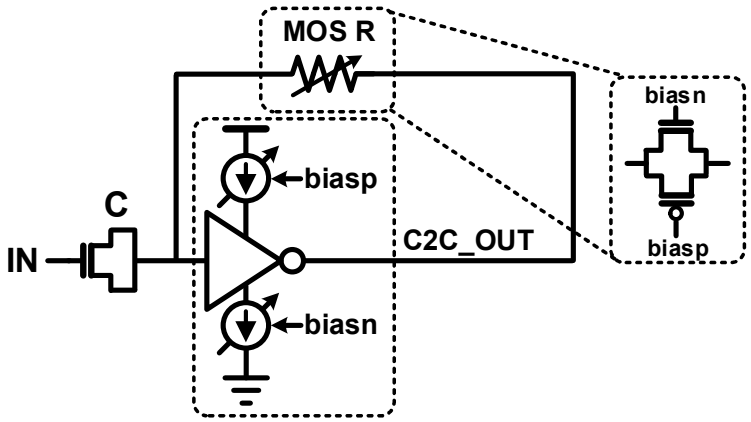


Fig. 5.12 Circuit of the proposed AC-coupling C2C for delay adjustment.

the feedback resistor controls the output swing, the delay in the C2C is adjusted by controlling the inverter current and the output swing. As VDD increases, the current decreases and the output swing increases, increasing the overall delay. On the other hand, as VDD decreases, the delay decreases because the current increases and the output swing decreases. However, this compensation method increases the overall delay because it is difficult to reduce the delay of the uncompensated inverter at low VDD to match the delay of the inverter at high VDD for drift compensation.

The generated bias voltage for voltage drift compensation deteriorates the effect of temperature drift. Figure 5.13 shows the temperature sensitivity of bias voltages generated in the bias generator. Since $|V_{th}|$ decreases with increasing temperature, the current in the bias generation circuit increases, which reduces biasn and increases biasp. When the temperature decreases, biasn increases and biasp decreases. In the inverter, the overdrive voltage (V_{ov}) increases due to the decrease in $|V_{th}|$, but the mobility degradation is slightly more dominant than V_{ov} due to the large V_{ov}. Therefore, the temperature sensitivity characteristic of the proposed bias generator makes the temperature drift worse than the inverter without voltage drift compensation.

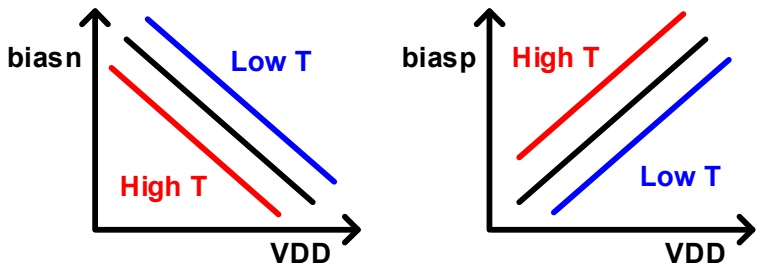


Fig. 5.13 Temperature sensitivity of biasn and biasp.

5.4 Circuit Implementation

The overall block diagram of the proposed clock distribution tree is shown in Figure 5.14. The distributed clock is divided into a CML path and a CMOS path. The global clock is distributed up to 400 μm distance as CML level. The CML path includes a 3-stage CML buffer, a CML divider, and a 6-stage CML buffer. The first stage, the 3-stage CML buffer, amplifies the 12 GHz CML level clock to ensure the CML divider operates correctly. The CML divider generates a 6 GHz CML quadrature clock and the 6-stage CML buffer drives the 400 μm wire. Afterward, every 100 μm, a CML quadrature clock is fed into the CMOS path. The CMOS path consists of a C2C, a 4-stage inverter, and a bias generation circuit for voltage drift compensation. The CMOS quadrature clocks are used on each DQ.

The architecture of CML buffers is the same as in Figure 5.1(a). The architecture of the D-flip-flop (DFF)-based CML frequency divider is shown in Figure 5.15. If there is only a DC component at the input of the divider, the divider self-oscillates at f_{self_osc}. Also, the divider has a sensitivity curve similar to the injection lock range of an injection lock frequency divider as shown in Figure 5.16 [30]. If the target output frequency of the divider is close to f_{self_osc}, the required minimum input swing of the divider for normal operation is very low. Therefore, the divider is implemented to have a f_{self_osc} at 7.52 GHz which is close to the target output frequency 6 GHz and slightly higher than 6 GHz.

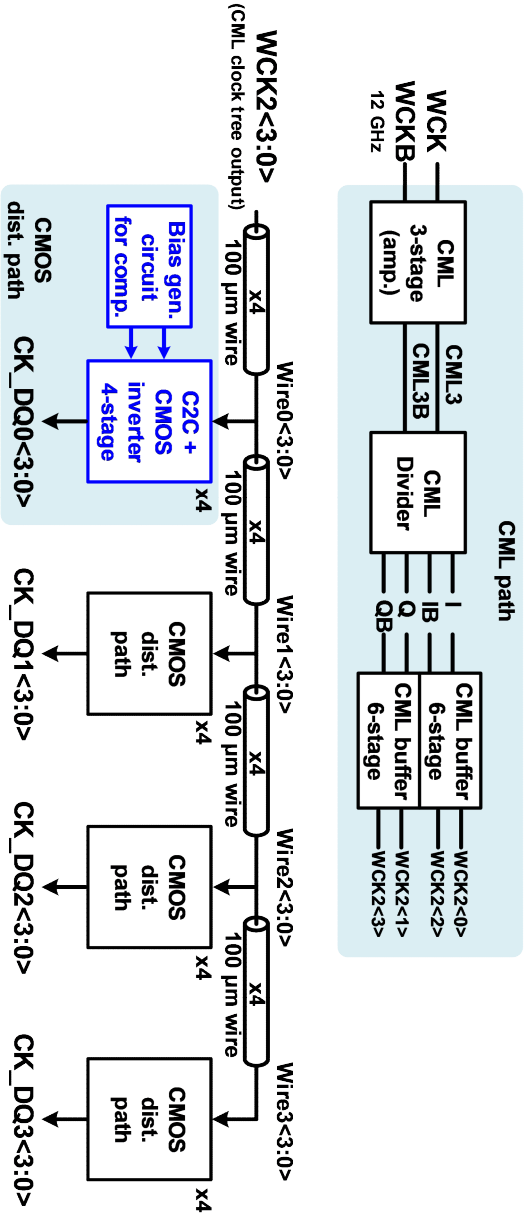


Fig. 5.14 Overall block diagram of proposed clock distribution tree.

The block diagram of the proposed CMOS path is shown in Figure 5.17. Both the current of the inverter and the feedback resistor are adjusted by biasn and biasp. Figure 5.18 shows a detailed C2C delay circuit in the proposed CMOS path. The current starved inverter is used and the feedback resistor consists of poly resistors and MOS R. In the inverter stage after the C2C, the current starved inverter is not used in the last stage for maintaining driving strength to drive the load capacitance.

The bias generator circuit for voltage drift compensation is shown in Figure 5.10. The simulation results of the supply voltage sensitivity of biasn and biasp are shown in Figure 5.19.

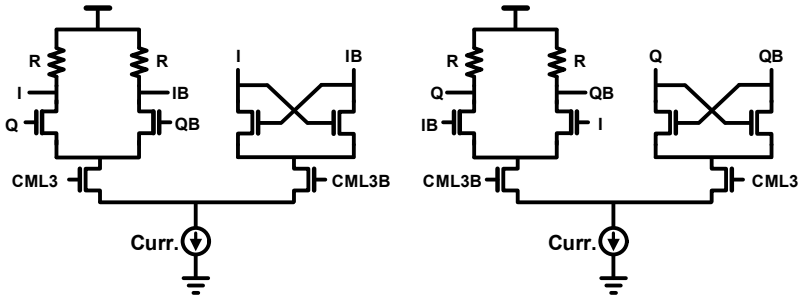


Fig. 5.15 DFF-based CML divider.

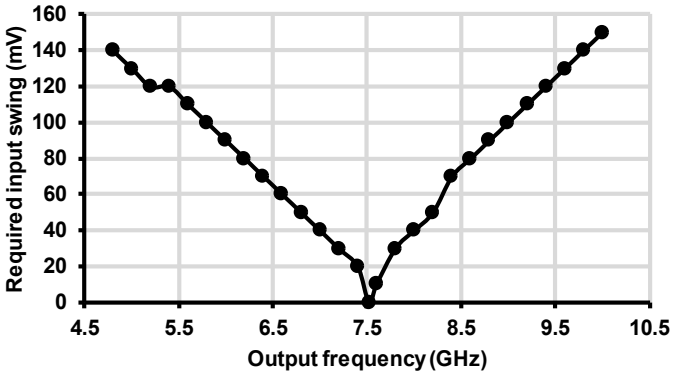


Fig. 5.16 Simulated sensitivity curve of CML divider.

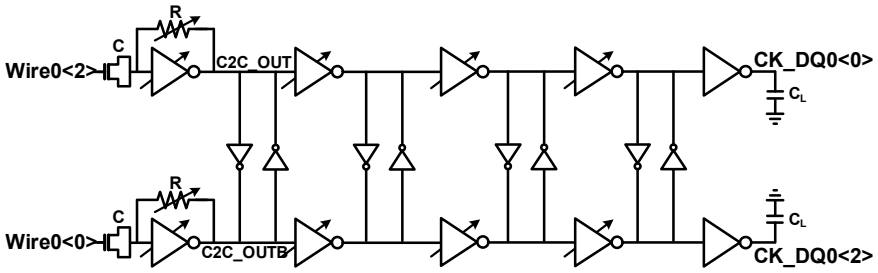


Fig. 5.17 Block diagram of the proposed CMOS path.

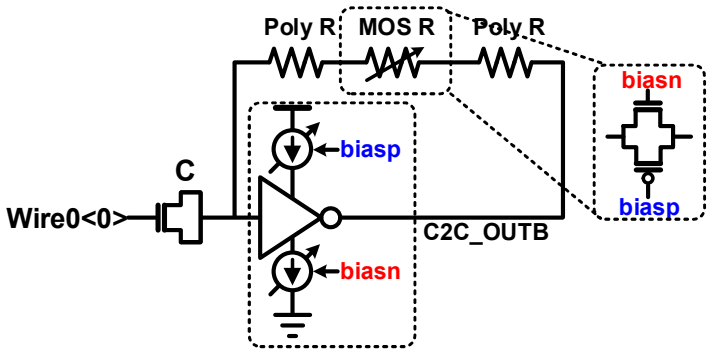


Fig. 5.18 C2C delay circuit in the proposed CMOS path.

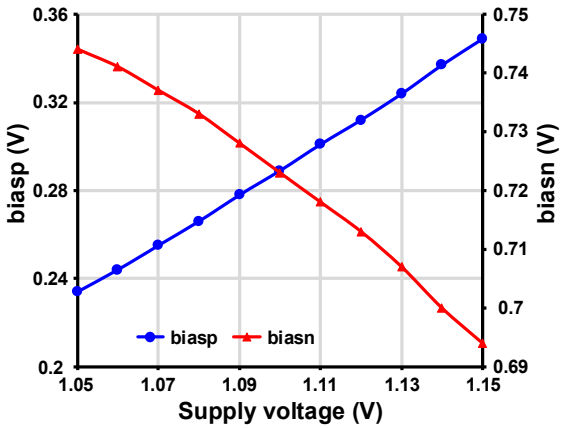
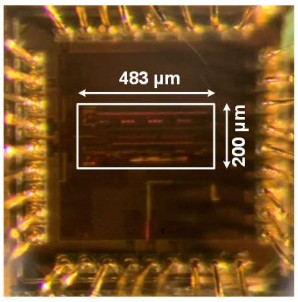


Fig. 5.19 Simulation results of biasn and biasp vs. supply voltage.

5.5 Measurement Results

The clock distribution tree applying the proposed scheme is fabricated in the 40-nm CMOS technology. The chip microphotograph and the core layout of the clock tree are shown in Figure 5.20. The conventional clock distribution is also implemented in the fabricated chip for performance comparison with the proposed scheme. The conventional CMOS path is composed of a conventional AC-coupling C2C and inverters. The bias generator occupies 0.0006 mm^2 and the area of the bias generator is large for controllability. The total area of the fabricated chip is 0.0966 mm^2.



	Block description	Area (µm²)
A	CML 3-stage	899.1
B	CML divider	308.1
C	CML 6-stage x2	1727.3
D	CMOS path (conventional) x2	545.4
E	CMOS path (proposed) x2	943
F	Bias generator	602.6

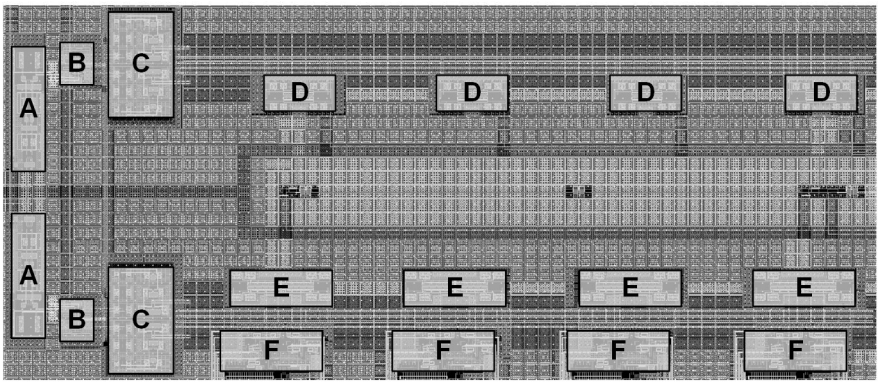


Fig. 5.20 Chip microphotograph and layout details of the clock distribution tree.

Figure 5.21 shows the measurement setup of the proposed scheme. MP1800A provides a 12 GHz differential clock to the chip. To compare the performance of the conventional scheme and the proposed scheme, the on-chip MUX selects one of the output clock of the conventional scheme and the output clock of the proposed scheme. After that, CLK_{OUT} is triggered by a 6 GHz trigger clock of MP1800A and jitter is measured on the oscilloscope. Also, the arbitrary function generator is used for supply voltage modulation.

At 6 GHz output clock and 1.1-V supply voltage, the power consumption of the conventional scheme is 8.67 mW and the power dissipation of the proposed scheme is 11.02 mW, which is amortized by four assuming 4 DQs.

The measured bias voltage is shown in Figure 5.22. As intended, biasn has negative supply sensitivity and biasp has positive supply sensitivity.

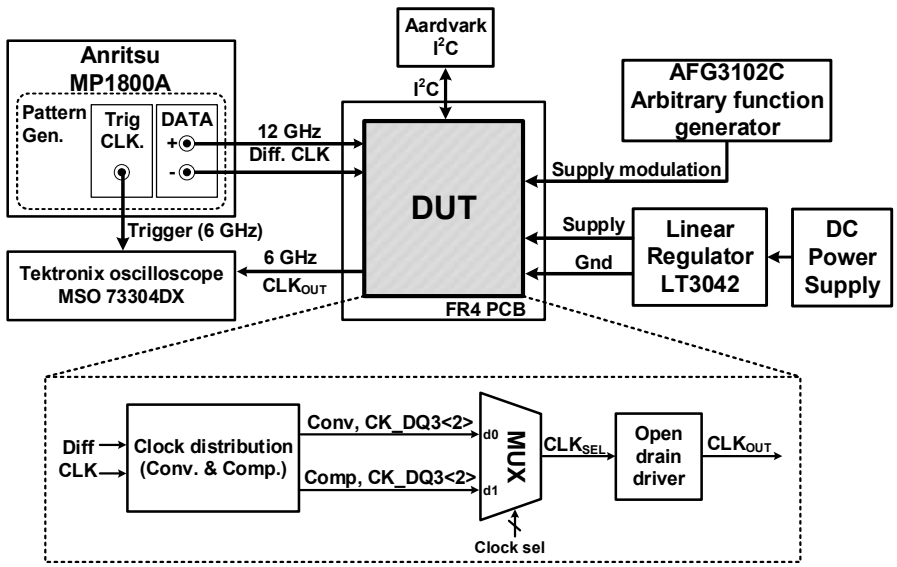


Fig. 5.21 Measurement setup.

The supply voltage is changed manually from 1.05 V to 1.15 V with 0.01 V step to verify voltage drift compensation. Afterwards, the delay from the trigger point to the output clock is measured. The measured delay versus the supply voltage is shown in Figure 5.23. The delay of the compensation scheme is larger than the conventional scheme due to voltage drift compensation, but peak-to-peak delay variation by supply voltage drift is reduced from 10.59 ps to 3.58 ps.

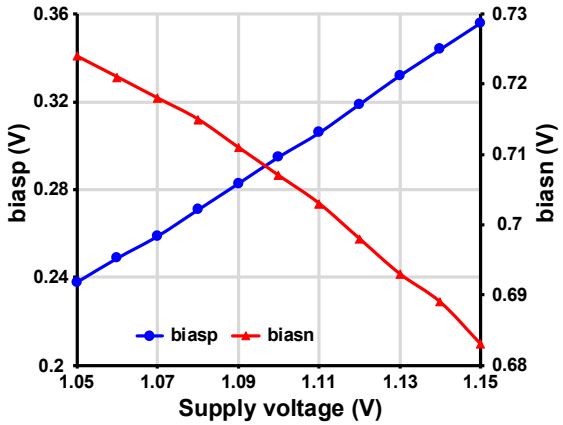


Fig. 5.22 Measured results of biasn and biasp vs. supply voltage.

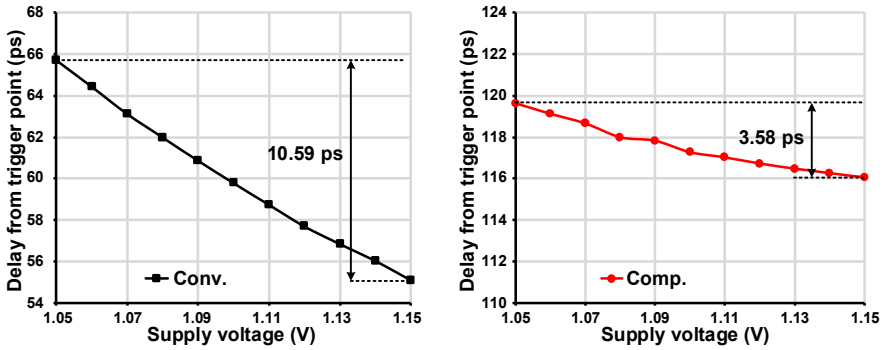
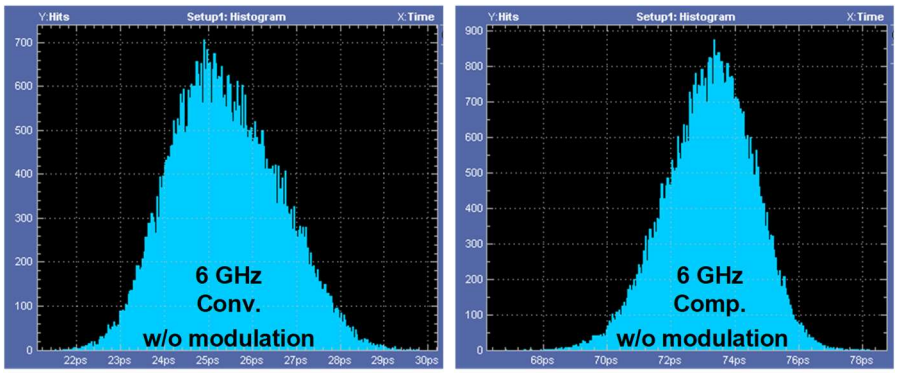


Fig. 5.23 Measured delay from trigger point vs. supply voltage.

The histogram of the measured jitter of the 6 GHz output clock at a 1.1 V supply voltage without supply voltage modulation is shown in Figure 5.24. The proposed scheme has larger jitter than the conventional scheme because the total delay of the clock distribution path increases due to voltage drift compensation.

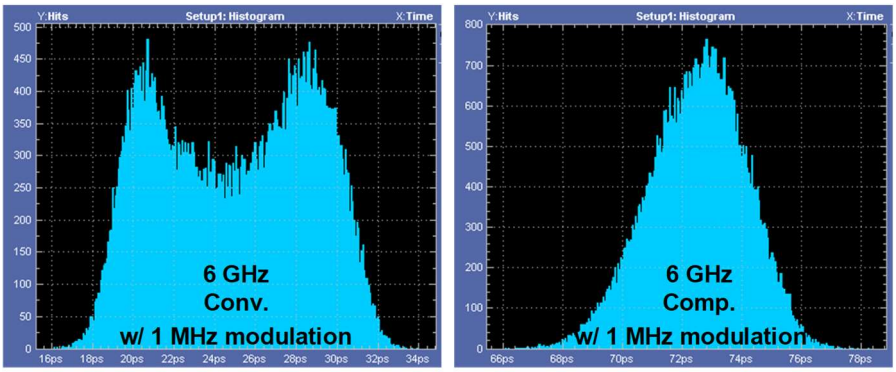
The measured jitter histogram when the 1 MHz, 100 mV_{P-P} sinusoidal wave modulates the supply voltage is shown in Figure 5.25. Since the proposed scheme compensates for voltage drift, there is only one peak in the jitter histogram of the proposed scheme compared to the conventional scheme which exhibits two peaks in the jitter histogram. Therefore, RMS jitter and P-P jitter of the proposed scheme are smaller than RMS jitter and P-P jitter of the conventional scheme.

The measured RMS jitter and P-P jitter versus the supply modulation frequency are shown in Figure 5.26.



	RMS jitter (ps_{RMS})	P-P jitter (ps_{P-P})
Conventional	1.21	8.29
Compensation	1.36	11.4

Fig. 5.24 Measured jitter histogram without supply voltage modulation.



	RMS jitter (ps_{RMS})	P-P jitter (ps_{P-P})
Conventional	3.77	17.85
Compensation	1.61	12.2

Fig. 5.25 Measured jitter histogram when 1 MHz, 100 mV_{P-P} sinusoidal wave modulates supply voltage.

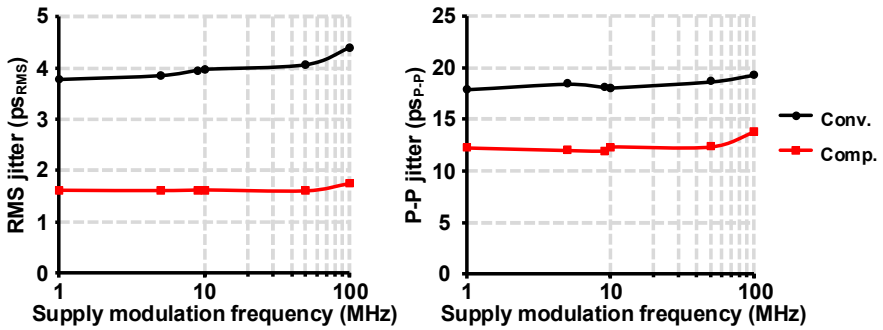


Fig. 5.26 Measured RMS jitter and P-P jitter vs. supply modulation frequency.

To check for jitter degradation due to temperature drift as mentioned in Section 5.3, jitter is measured when temperature drifts from 18 °C to 50 °C and the supply voltage is 1.1 V. Figure 5.27 shows measured jitter histogram when temperature drifts. RMS jitter of the conventional scheme and the proposed scheme is 2.68 ps_{RMS} and 2.7 ps_{RMS}, respectively. As expected, jitter increases, but the amount of increased jitter is negligible.

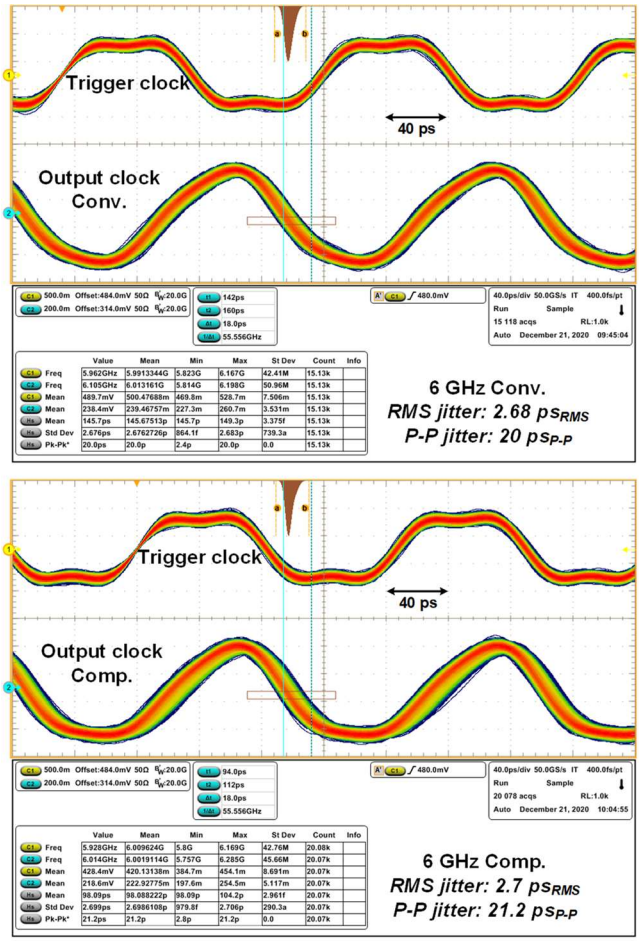


Fig. 5.27 Measured jitter histogram when temperature drifts.

Table 5.1 shows the performance summary of the proposed voltage drift compensation scheme. Using the proposed open-loop based scheme reduces the induced jitter due to voltage drift while minimizing power and area overhead.

Table 5.1 Performance summary

	JSSC 03' [5]	JSSC 07' [31]	This work	
Technology	250 nm	130 nm	40 nm	
Supply voltage (V)	2.5	1	1.1	
Frequency (GHz)	1	1.4	6	
			Conv.	Comp.
Area (mm^2)	0.028	0.064	0.003*	0.004*
Power consumption (mW)	10	9.6	8.67**	11.02**
RMS jitter without supply modulation (ps_{RMS})	3.28	3.9	1.21	1.36
Amplitude of supply noise	N/A	10 mV	50 mV	
RMS jitter with 1 MHz supply modulation (ps_{RMS})	N/A	3.95	3.77	1.61
RMS jitter with 10 MHz supply modulation (ps_{RMS})	N/A	3.97	3.97	1.62
Normalized increased amount of RMS jitter due to supply modulation***	N/A	0.06 (1 MHz) 0.07 (10 MHz)	0.07 (1 MHz) 0.08 (10 MHz)	0.02 (1 MHz) 0.02 (10 MHz)
Power overhead	30 %	N/A	-	27.1 %
Area overhead	50 %	25.5 %	-	28.7 %

*only includes DQ0<3:0> path (conv.: A+B+C+D, comp.: A+B+C+E+F)
**amortized by 4 (assume 4 DQs)
***Increased amount of RMS jitter (ps_{RMS})/noise amplitude (mV)

Chapter 6

Quadrature Error Corrector with Minimum Total Delay Tracking

6.1 Overview

In a half-rate system, DQ is synchronized to a differential clock whose frequency is half the data rate as shown in Figure 6.1. If the duty cycle (D) of the clock is not 0.5, the data valid window (tDV) decreases. Therefore, the duty cycle correction is necessary. As data transfer rates increase, clock frequencies used for high-speed data paths also increase. Thus, multi-phase clocks are typically utilized in DRAM to relax timing margins because of the reduced timing budget. Figure 6.2 shows the timing diagram of the quarter-rate system. When the time difference between I and Q (Δt_{I-Q}) is a quarter of the clock period (T/4), tDV is the maximum. However, if Δt_{IB-}

$_{QB}$ is deviated from T/4, tDV decreases. Phase skew between multi-phase clocks is occurred due to device and layout mismatch. Since clocks are located in the middle of the chip and distributed, distributed clocks may experience different line lengths, which causes phase skew. To reduce phase error, multi-phase correction is required.

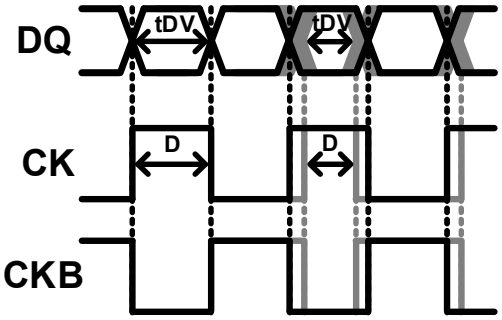


Fig. 6.1 Timing diagram of half-rate system.

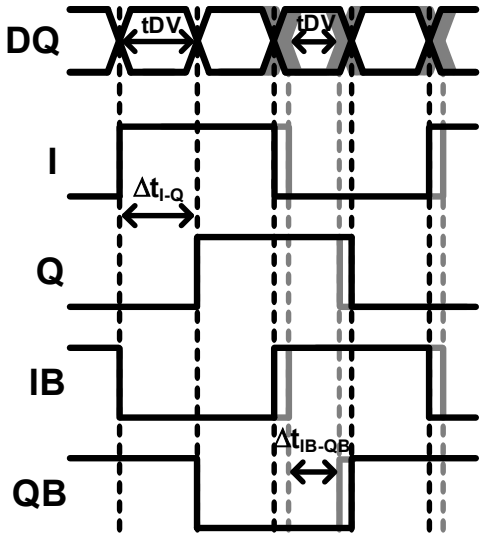


Fig. 6.2 Timing diagram of quarter-rate system.

6.2 Prior Works

Several multi-phase correction schemes have been proposed. The active poly-phase filter-based open-loop scheme exhibits a small RMS jitter contribution but remains the phase error after the error correction [7]. Furthermore, due to the injection-locking characteristic, it requires a few cascading stages for a wide-range operation. Figure 6.3 shows a distributed DLL-based multi-phase generator [8]. The control loop includes a phase detector (PD), a voltage-to-current converter (V/I), and a capacitor as a loop filter (LF). Each control loop adjusts each delay cell to generate accurate multi-phase clocks. The distributed DLL offers a small RMS jitter but the residual phase error is non-negligible as well due to the mismatch of error detection circuits in each control loop. The phase error corrector with a relaxation oscillator-based phase detector is also susceptible to the mismatch [9].

The digital DLL-based scheme adopts a shared digital feedback loop to eliminate the effect of mismatch as shown Figure 6.4 [10]. It updates four delay cells (t_Q, t_{IB},

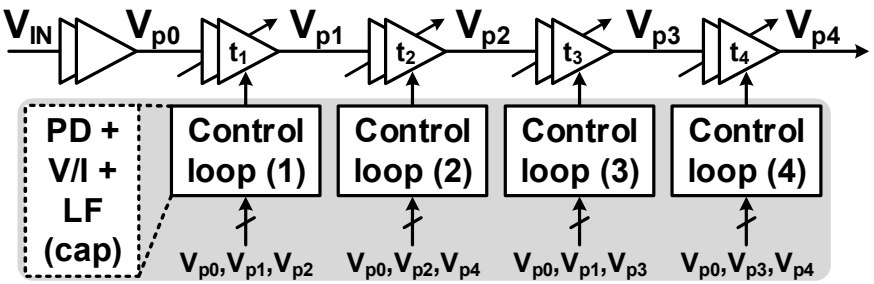


Fig. 6.3 Distributed DLL-based multi-phase generator.

t_{QB}, t_{quad}) sequentially to make the time difference between adjacent clocks t_{quad}. As a result, t_{quad} is equal to a quarter of the clock period (T/4) and 4 phase error is removed. However, it shows a larger RMS jitter contribution than the distributed DLL due to quantization noise and the increased clock path delay. Since the delay of in-phase clock (t_I) is fixed at mid-point by setting the code of DCDL, the overall set of codes of DCDLs may not be at the optimum in terms of jitter, which leads to degradation of the data eye. For instance, in the case of Figure 6.5, there are many solutions to correct quadrature errors. Assuming I_{IN} is later than its ideal position by 2 and t_I is fixed at 7, t_Q, t_{IB}, and t_{QB} will be 9 because I_{IN} is later than its ideal position and other phases are ideal ($\Delta t_{I\text{-}Q} < T/4$, $\Delta t_{QB\text{-}I} > T/4$, and other time differences of adjacent clocks are T/4). In this case, the increased delay is greater than required to remove phase error. On the other hand, if t_I is 0, t_Q, t_{IB}, and t_{QB} are 2. This is the best solution since the sum of the increased delay (= $t_I + t_Q + t_{IB} + t_{QB}$) in the clock path is the minimum. Therefore, to minimize the total delay, at least one of the delay in the

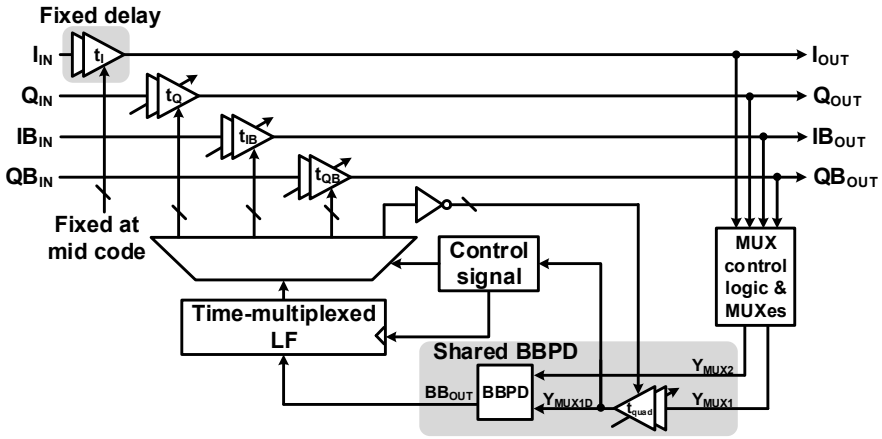


Fig. 6.4 Digital DLL-based 4 phase corrector.

quadrature clock path should be at the minimum.

In this paper, an improved quadrature error corrector (QEC) which adopts the shared feedback loop of [10] is proposed [12]. The calibration loop of the scheme finds the solution that minimizes the total delay in the clock path. In addition, an asynchronous and seamless calibration on-off scheme is also suggested for the reduction of power consumption in the idle state after calibration.

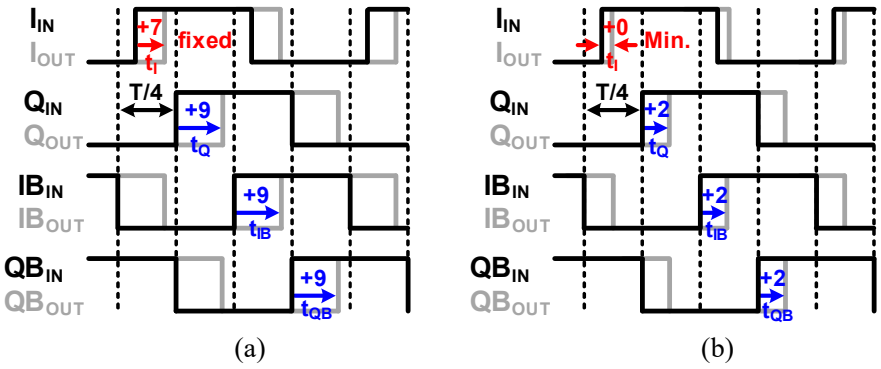


Fig. 6.5 Timing diagram of phase error corrected output clocks when the solution for error correction is (a) not optimal and (b) optimal.

6.3 Quadrature Error Correction Method

Figure 6.6 shows the conceptual block diagram of the proposed QEC. The signal selector selects two adjacent clocks with 90° apart. The selection of the two phases is rotated over 4 clocks in a continuously running loop. For example, the output sequence of the signal selector is (I,Q) → (Q,IB) → (IB,QB) → (QB,I).

The quadrature error detector (QED) detects the phase skew between selected clocks as shown Figure 6.7(a). The preceding one of the selected clocks (O_{MUX0}) is delayed in the DCDL by t_{quad0} that is adjusted by the control signal C_{QUAD} and the other clock (O_{MUX1}) is delayed in another DCDL by t_{quad1} with the minimum delay

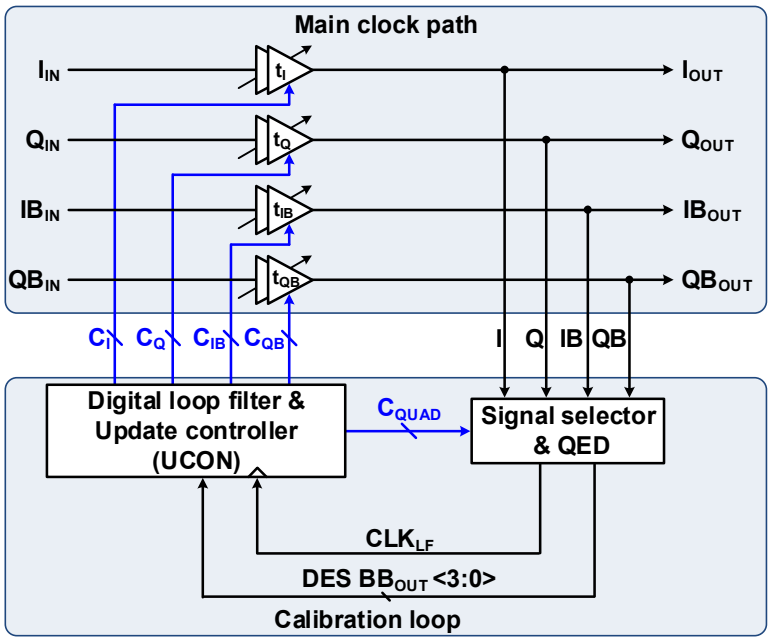


Fig. 6.6 Conceptual block diagram of proposed QEC.

code. The Bang-Bang PD (BBPD) compares the two delayed clocks O_{MUX0D} and O_{MUX1D} to collect quadrature error information. Figure 6.7(b) presents the case when I and Q are chosen. If O_{MUX0D} leads O_{MUX1D}, the output of BBPD (BB_{OUT}) is '0'. Otherwise, BB_{OUT} is '1'. The phase comparison is repeated over 4 pairs of adjacent clocks and the BBPD outputs are deserialized to a 4-bit wide DES BB_{OUT}. The DLF filters DES BB_{OUT}. Afterwards, the five DCDL codes, C_I, C_Q, C_{IB}, C_{QB}, and C_{QUAD} are updated by the update controller (UCON) to align O_{MUX0D} and O_{MUX1D}. Therefore, the time difference between adjacent clocks will be the same as t_{quad}, which is the time difference between t_{quad0} and t_{quad1}. As a result, at the locked state, all adja-

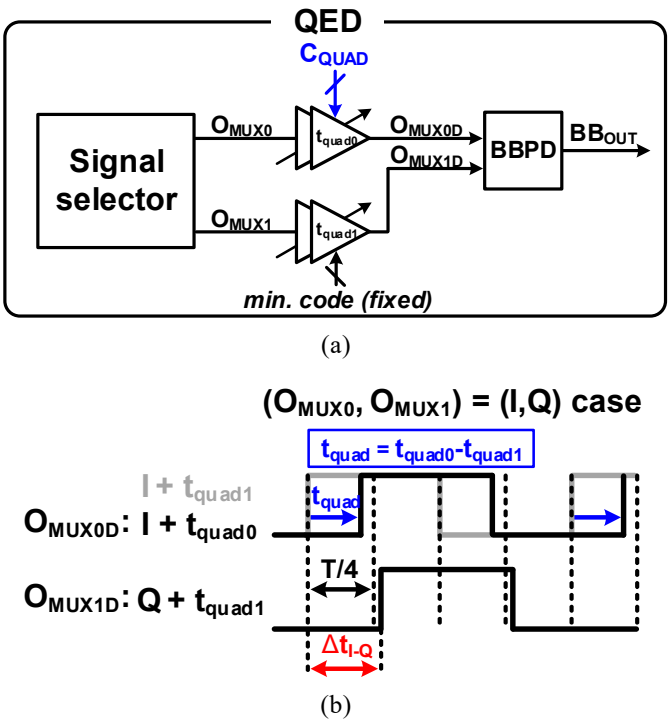


Fig. 6.7 (a) Block diagram and (b) timing diagram of QED.

cent time differences will be t_{quad} as expressed in 6.1. 6.2 shows the sum of time differences is equal to the clock period. Consequently, the adjacent time difference is a quarter of the clock period as shown in 6.3.

$$\Delta t_{I-Q} = \Delta t_{Q-IB} = \Delta t_{IB-QB} = \Delta t_{QB-I} = t_{quad} \quad (6.1)$$

$$\Delta t_{I-Q} + \Delta t_{Q-IB} + \Delta t_{IB-QB} + \Delta t_{QB-I} = T \quad (6.2)$$

$$\Delta t_{I-Q} = \Delta t_{Q-IB} = \Delta t_{IB-QB} = \Delta t_{QB-I} = T/4 \quad (6.3)$$

The phase errors are removed when the time difference of all adjacent clocks is locked at t_{quad}=T/4. Thus, not only the delay control signals of the four clock paths, but also the delay control C_{QUAD} of the quadrature delay line must be determined. Figure 6.8 illustrates the timing diagram of O_{MUX0} and O_{MUX1D} and the update direction of each delay cell to align two delayed signals. In case (1), to align O_{MUX0D} and O_{MUX1D}, t_I and t_{quad0} should be increased and t_Q should be decreased. In case (2), t_Q and t_{quad0} should be decreased and t_{IB} should be increased. Thus, taking into account the results of two comparisons, the update of t_{quad0} is canceled out. Updating 3 delay

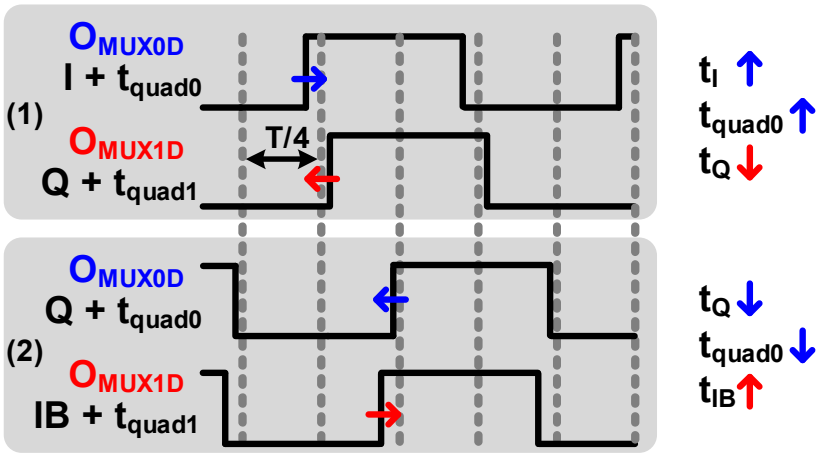


Fig. 6.8 Timing diagram and update direction of each delay cell to align O_{MUX0D} and O_{MUX1D}.

cells corresponding to the one phase comparison result requires 3 adders, but 3 adders is unnecessary due to cancellation, so 4 BB_{OUT}s are collected and only one DCDL is adjusted to reduce hardware complexity.

There are several candidates of DCDL whose update polarity is up or down as a result of gathering BB_{OUT}s. Furthermore, four control codes for the DCDLs in the main clock path do not have a unique solution to remove phase error because there are many combinations that can eliminate the phase skews. To solve this problem, an internal state of the DCDL update logic, update direction state (UDS) is introduced to decide which DCDL to update in terms of minimizing the total increased delay. Consequently, UCON combines the 4-bit wide DES BB_{OUT} and UDS to choose only one DCDL control register to update and how to update, up or down to reach the optimum solution as shown in Figure 6.9.

Figure 6.10 presents a flow chart of UDS. Before updating the DCDL, check if at least one of C_I, C_Q, C_{IB}, and C_{QB} is 0 because that indicates the minimum code and is a necessary condition to minimize the total increased delay. Without this condition, many solutions exist, causing wandering of DCDL codes. Therefore, due to this constraint, at least one of the main DCDL codes should contain the minimum code and such code is the unique solution that eliminates quadrature error. If all codes are

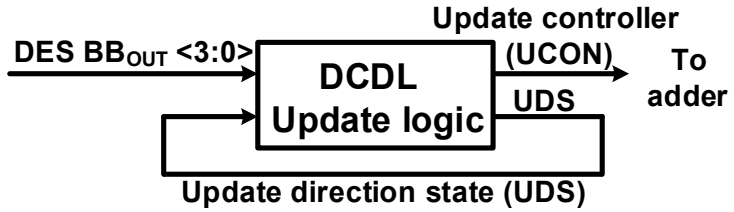


Fig. 6.9 Block diagram of DCDL update logic.

larger than 0, UDS will be DN to decrease the codes of DCDL for reducing the delay. Otherwise, UCON selects the DCDL and the update polarity by the combination of DES BB_{OUT} and UDS. After that, an adder in the DLF calculates the corresponding code of DCDL. If underflow occurs in the calculation, the calculated code is not updated and UDS will be reversed as UP to search other solution which increases the codes of DCDL.

On the other hand, the DCDL code is updated by the adder.

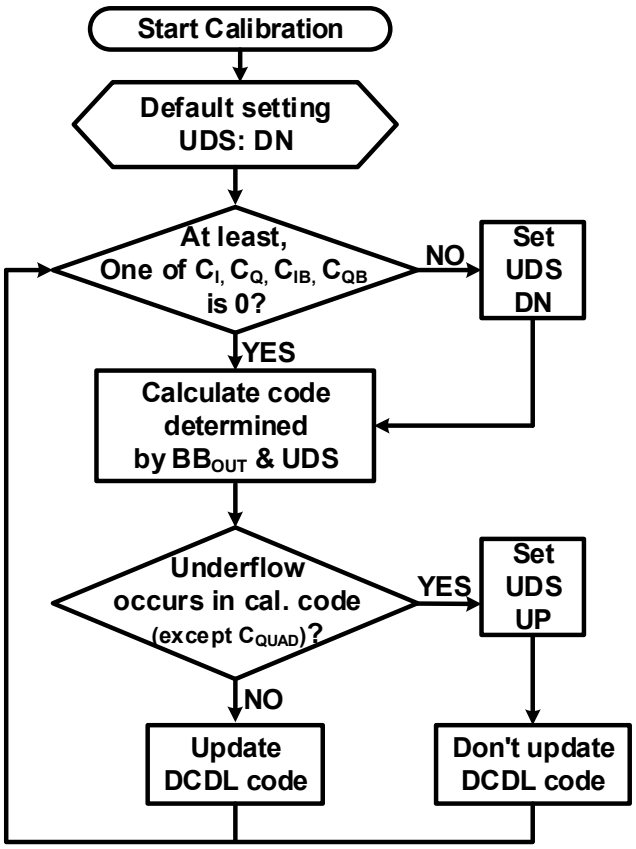
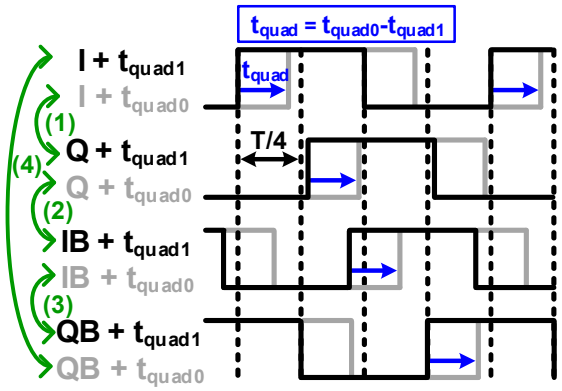


Fig. 6.10 A flow chart of UDS.

There are 3 examples of the proposed QEC. As shown in Figure 6.11, where a one-bit outcome of BB_{OUT} informs that three DCDLs (two main DCDLs and quadrature DCDL) should be controlled to align O_{MUX0D} and O_{MUX1D}, 4-bit BB_{OUT}s and UDS are combined to select only one of the four main DCDLs or the quadrature DCDL. In this case, reducing t_Q or increasing t_{IB} would be the solution, but owing to the UDS, only one DCDL code is selected. If the current value of the UDS is DN then decreasing t_Q is selected or increasing t_{IB} is chosen.

Figure 6.12 illustrates the example when the quadrature DCDL (C_{QUAD}) is selected. When all BB_{OUT}s are 0, the updates of main DCDLs are canceled from each other. Therefore, C_{QUAD} is chosen only if 4-bit wide BB_{OUT} suggests to control C_{QUAD} in the same direction. In Figure 6.11, the selection of the main DCDL is determined by



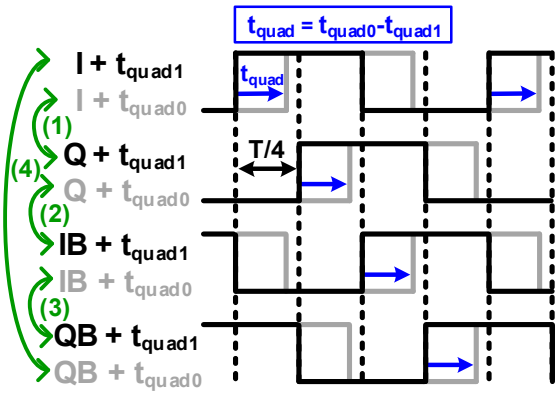
Comparison pair	BB_{OUT}	Candidates to update
(1)	0	$t_I \uparrow$, $t_Q \downarrow$, $t_{quad0} \uparrow$
(2)	1	$t_Q \downarrow$, $t_{IB} \uparrow$, $t_{quad0} \downarrow$
(3)	0	$t_{IB} \uparrow$, $t_{QB} \downarrow$, $t_{quad0} \uparrow$
(4)	0	$t_{QB} \uparrow$, $t_I \downarrow$, $t_{quad0} \uparrow$

DES BB_{OUT} <3:0>	UDS	UCON
0100	DN	$t_Q \downarrow$
	UP	$t_{IB} \uparrow$

Fig. 6.11 Example: when the DCDL of the main clock path is selected.

the UDS, which is to keep the overall total main clock path delays at the minimum. However, the UDS does not affect the update polarity of C_{QUAD} because only C_{QUAD} remains as a candidate and all update polarities are upward if all BB_{OUT}s are 0.

If 4-bit BB_{OUT} is 0101 or 1010, the number of the selectable DCDLs can be more than one even given UDS as shown in Figure 6.13. Given the UDS, there are two candidates. In the proposed scheme, decreasing t_{QB} or increasing t_{IB} is decided as an output of UCON if UDS is DN or UP, respectively. Regardless of which DCDL is selected, the 4-bit BB_{OUT} changes and transitions to another state by adjusting delay. As a result, the DCDL codes converge to remove phase skew and one of the main DCDL codes is always at '0', thereby added jitter is minimized thanks to the shortest total delays in the four clock paths.



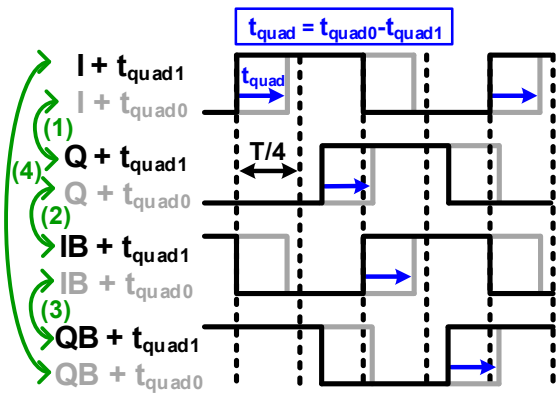
Comparison pair	BB_{OUT}	Candidates to update
(1)	0	$t_I \uparrow, t_Q \downarrow, t_{quad0} \uparrow$
(2)	0	$t_Q \uparrow, t_{IB} \downarrow, t_{quad0} \uparrow$
(3)	0	$t_{IB} \uparrow, t_{QB} \downarrow, t_{quad0} \uparrow$
(4)	0	$t_{QB} \uparrow, t_I \downarrow, t_{quad0} \uparrow$

DES BB_{OUT} <3:0>	UDS	UCON
0000	DN	$t_{quad0} \uparrow$
	UP	$t_{quad0} \uparrow$

Fig. 6.12 Example: when quadrature DCDL is selected.

An asynchronous calibration on-off mode is supported to reduce the power consumption after the skew correction is accomplished. In the on mode, the calibration loop works to correct phase errors. In the off mode, DCDL codes are frozen until re-activated and the calibration loop is turned off by clock gating of the signal selector and the DLF. When an external signal, CAL, goes '0' which might come asynchronously at any time with respect to the incoming clocks, the calibration loop is deactivated and keeps its state by two control signals, EN_A and EN_B, generated in the enable sequencer (ENS) using synchronizers and clock gating. When re-activated on CAL going '1', the previous state is recovered and the calibration starts seamlessly.

The block diagram and the timing diagram of the calibration on-off scheme are shown in Figure 6.14. EN_A is used as clock gating for the DLF clock CLK_{LF} and



Comparison pair	BB_{OUT}	Candidates to update
(1)	0	$t_I \uparrow, t_Q \downarrow, t_{quad0} \uparrow$
(2)	1	$t_Q \downarrow, t_{IB} \uparrow, t_{quad0} \downarrow$
(3)	0	$t_{IB} \uparrow, t_{QB} \downarrow, t_{quad0} \uparrow$
(4)	1	$t_{QB} \downarrow, t_I \uparrow, t_{quad0} \downarrow$

DES BB_{OUT} <3:0>	UDS	UCON
0101	DN	$t_Q \downarrow, t_{QB} \downarrow$
	UP	$t_I \uparrow, t_{IB} \uparrow$

Fig. 6.13 Example: when the number of the selectable DCDLs is more than one.

EN_B controls the inputs of the signal selector to turn on or off the calibration loop. The suitable on-off sequence of enable signals is important to avoid malfunction or deadlock due to metastability of the samplers. When CAL changes from off to on, EN_B becomes high before EN_A in order to enable CLK_{LF} only after the inputs of the signal selector are settled. On the other hand, when CAL changes from on to off, EN_A enters into off before EN_B to disable CLK_{LF} first to avoid accidentally sampling the wrong input to the DLF. Therefore, the ON→OFF sequence is (1)→(2)→(3) and the OFF→ON sequence is (4)→(5)→(6).

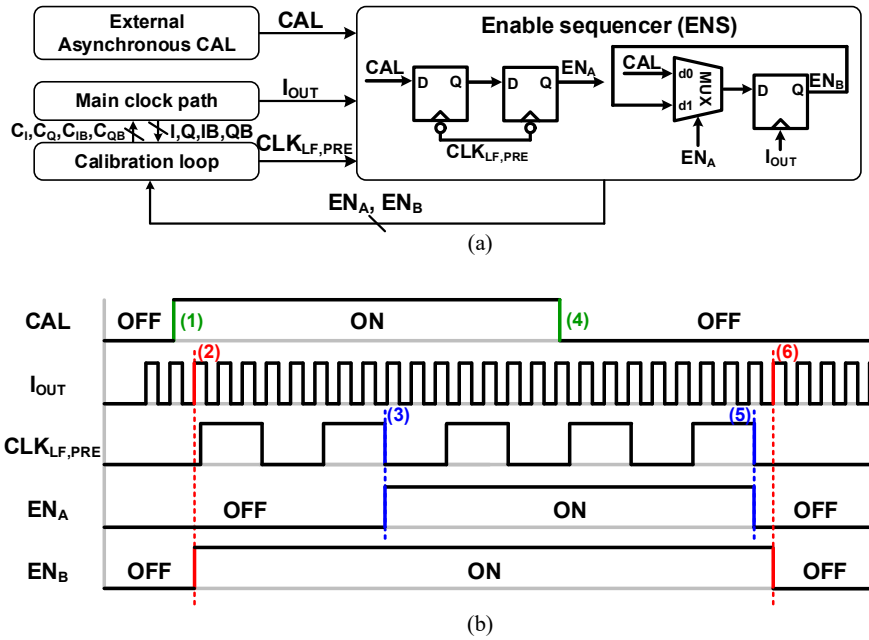


Fig. 6.14 (a) Block diagram and (b) timing diagram of asynchronous calibration on-off scheme.

6.4 Circuit Implementation

The overall architecture of the proposed QEC is shown in Figure 6.15. The QEC consists of a main clock path with four DCDLs, the ENS for the asynchronous calibration on-off, and the calibration loop. ENS generates EN_A and EN_B to turn the calibration loop on or off by an external signal CAL. The calibration loop is composed of the signal selector, the QED, and the DLF including the UCON. The delays of DCDLs in the main clock paths, t_I, t_Q, t_{IB}, and t_{QB} are adjusted to remove the phase error at the output. The signal selector chooses two adjacent clocks and the QED

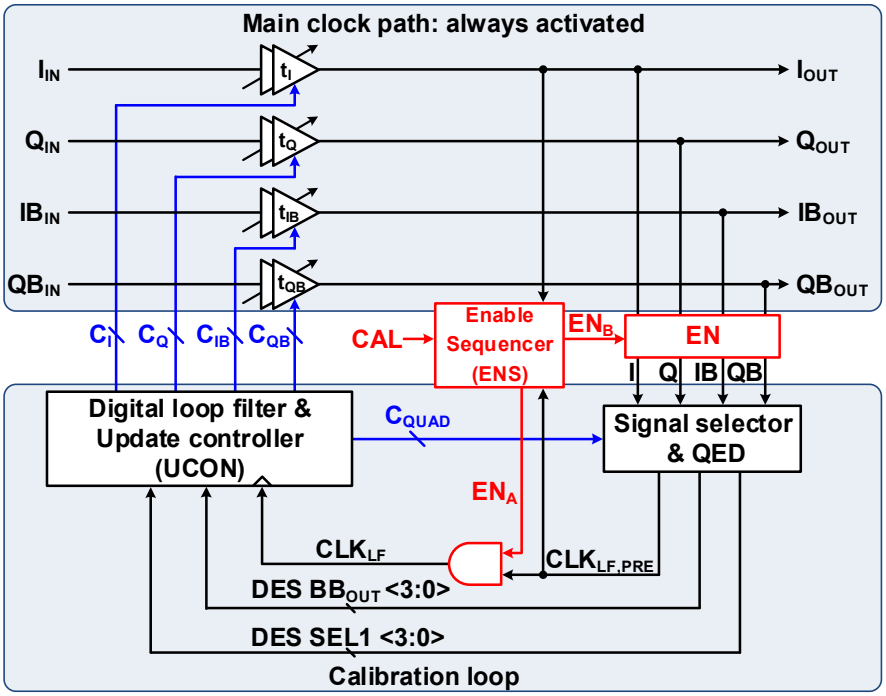


Fig. 6.15 Overall architecture of the proposed quadrature error corrector.

detects phase skew between the selected clocks. SEL1 of the signal generator is deserialized to indicate the sequence of the selected clocks because the sequence can change with each measurement if the counter is not reset. Afterwards, the UCON uses the information from DES BB_{OUT} to update one of five delay cells.

Figure 6.16 presents an enable circuit (EN) consisting of an AND gate. If EN_B is '1', one of EN's outputs, I, outputs the same phase as I_{OUT}. Otherwise, I will be 0 to

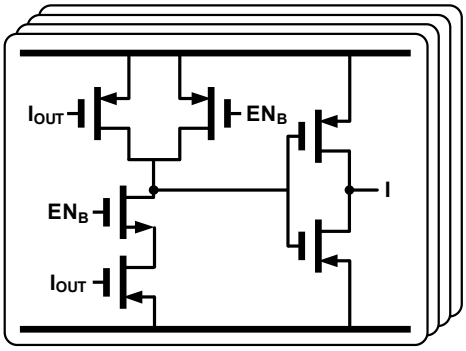


Fig. 6.16 Enable circuit (EN): AND gate.

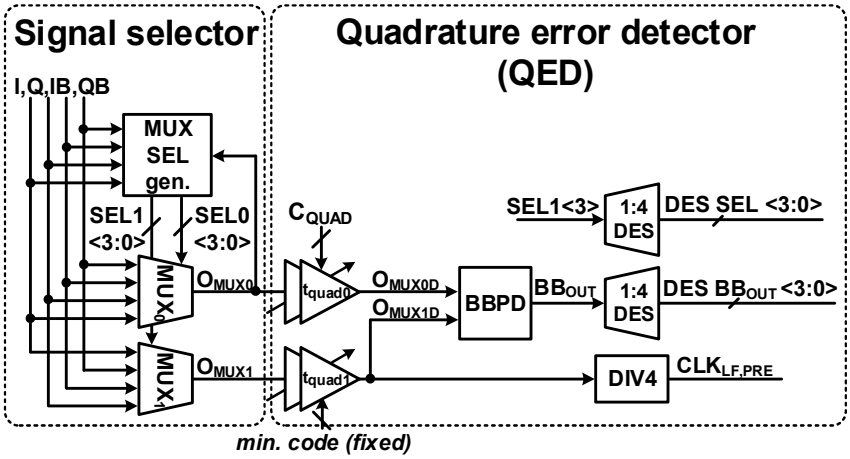


Fig. 6.17 Block diagram of the signal selector and QED.

disable the signal selector and QED. Figure 6.17 shows the block diagram of the signal selector and QED. The signal selector includes two multiplexers (MUXes) and the MUX SEL generator. The SEL0 controls the MUX_0 output and the SEL1 controls the MUX_1 output.

The block diagram and the timing diagram of the MUX SEL generator is shown in Figure 6.18. 2-bit binary counter is synchronized with O_{MUX0}, the output of MUX_0. After passing through the one-cold decoder and a NOR gate, D<3:0> is retimed by the negative edge of clocks to generate the SEL0 and the SEL1. In order to achieve the timing margin of T/2, to keep the level constantly low while the phase switching occurs, and to prevent glitches on O_{MUX0} and O_{MUX1}, SELs are created by retiming on the negative edge of clocks. The SEL0 and SEL1 are 4-bit one-hot coded to remove the possibility of being at the wrong state at power-up. If the least significant

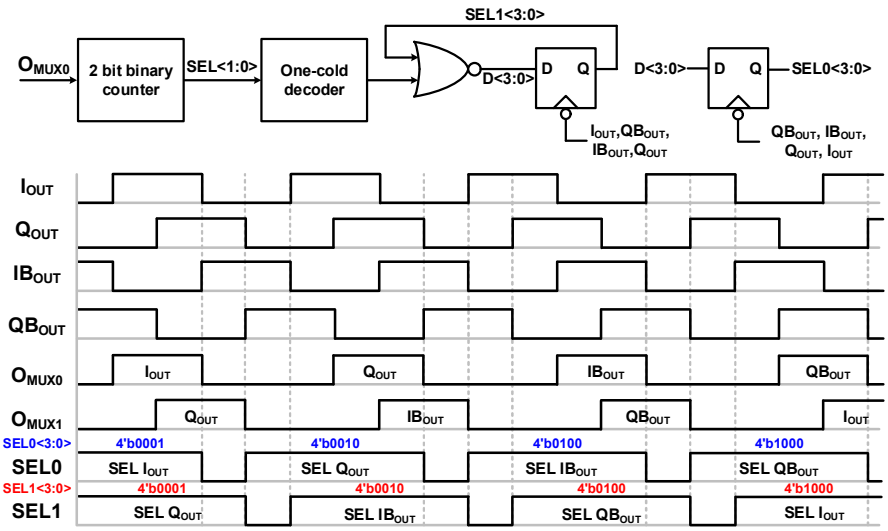


Fig. 6.18 The block diagram and the timing diagram of the MUX SEL generator.

bit of the SEL0 and the SEL1 is 1, O_{MUX0} outputs I_{OUT}, and O_{MUX1} outputs Q_{OUT} because O_{MUX0} must be the leading clock.

In the QED, the BBPD compares two delayed clocks and SEL1<3> and BB_{OUT} are deserialized. In addition, $CLK_{LF,PRE}$ is generated by dividing O_{MUX1D}. The frequency of O_{MUX1D} becomes 0.8 times the clock frequency (f), so the frequency of $CLK_{LF,PRE}$ become 0.2 times f.

Figure 6.19 shows the block diagram and the measured delay curve of the main DCDL. To cover the wide error correction range, the main DCDL employs NAND lattice-based dual coarse delay lines [32] merged with a PI based on tri-state inverters. Since the resolution of the DCDL affects the accuracy of the skew correction, a

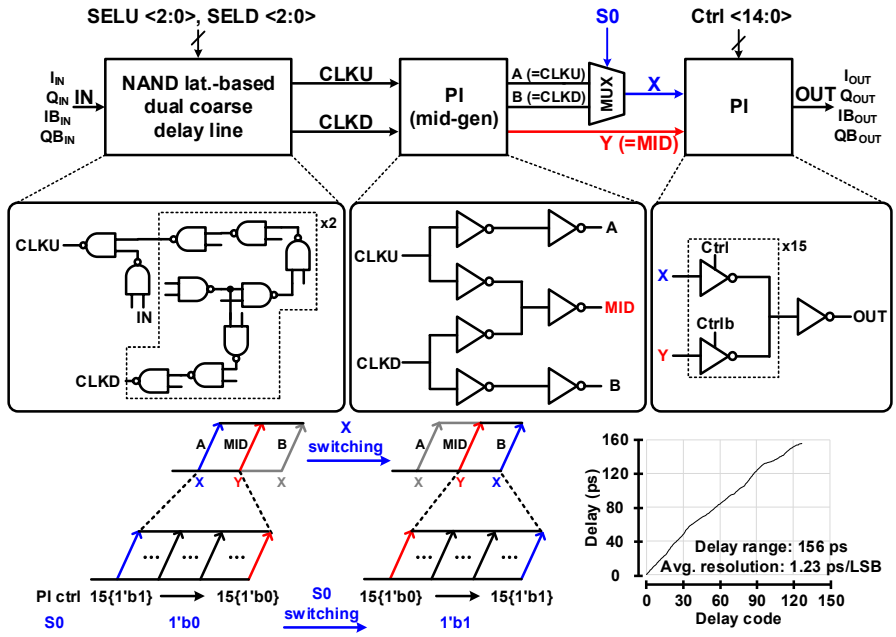


Fig. 6.19 The block diagram and the measured delay curve of DCDL in the main clock path.

fine resolution of $t_{NAND}/15$ is achieved using a pipelined interpolator [33], which consists of the cascaded mid-phase generator (mid-gen) PI and the MUX. Moreover, in the proposed DCDL, one of 15-bit controlled PI inputs is set at MID (Y) for seamless boundary switching in order to avoid large jitter jump when codes of main DCDLs are located at the boundary. The measured adjustable delay range is 156 ps and the average resolution is 1.23 ps/LSB.

The timing diagram of the DLF operation is illustrated in Figure 6.20. To avoid large dithering due to loop delay [34], MODE, a counter that is synchronized with

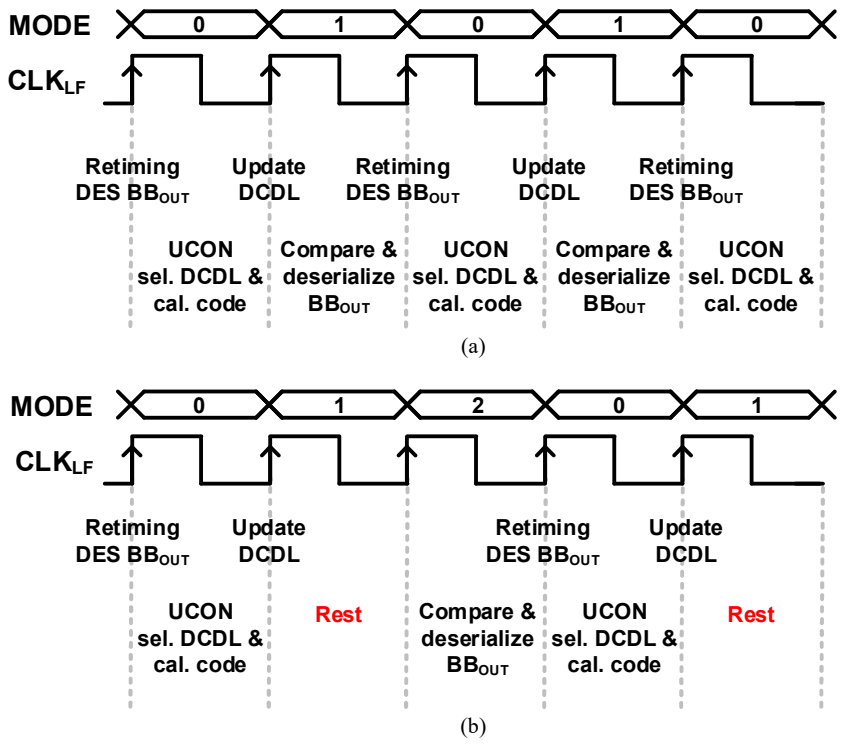


Fig. 6.20 Timing diagram of the DLF operation (a) without rest cycle and (b) with rest cycle.

CLK_{LF}, is introduced. When MODE is 0, the DLF retimes DES BB_{OUT} and the UCON selects DCDL and calculates the code of the selected DCDL. When MODE is 1, the calculated code is updated, followed by phase comparison and deserialization of the comparison results in Figure 6.20(a). However, if the phase of the two delayed clocks is compared immediately after the DCDL update, limit cycle might be encountered due to settling time of the DCDL code transition. If the value of BB_{OUT} is the value that was compared before the code update, DES BB_{OUT} notifies DLF of incorrect information. Therefore, the rest period, which is one cycle of CLK_{LF} after DCDL update, is introduced as shown in Figure 6.20(b). It increases the locking time of the proposed QEC but avoids large jitter caused by the limit cycle.

6.5 Measurement Results

The proposed QEC was fabricated in the 40 nm CMOS and the active area is 0.0428 mm^2 as shown in Figure 6.21. Figure 6.22 shows the measurement setup of the proposed scheme. In the test chip, four clock phases are generated internally with a 1.6 GHz to 4.6 GHz differential clock by an IQ divider. Moreover, programmable delay cells are added to the paths of the quadrature clock inputs to intentionally add arbitrary input phase skews. Therefore, the performance of the QEC can be measured in various input skew situations. If four output clocks are monitored externally at the same time, an additional phase error will occur due to mismatch in

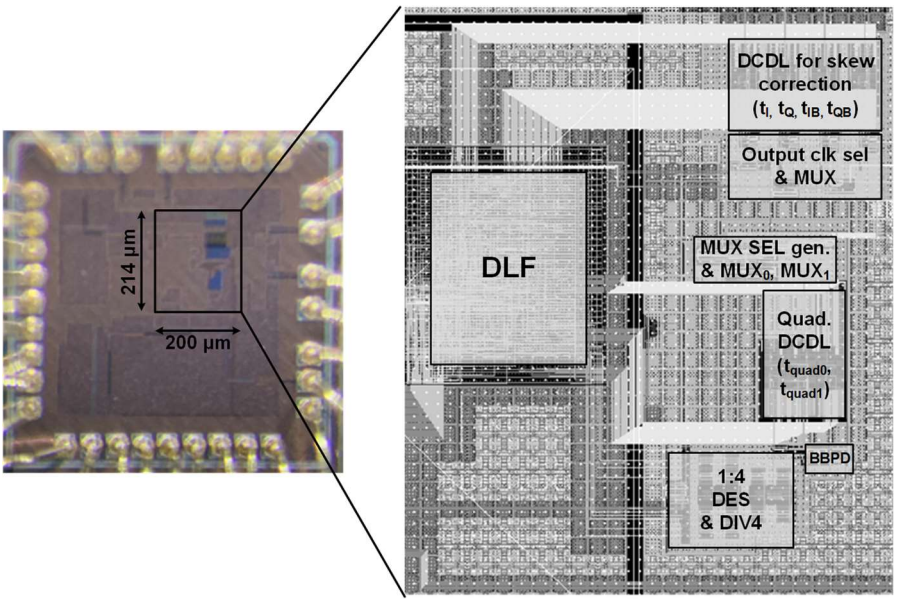


Fig. 6.21 Chip microphotograph and layout details of the proposed QEC.

cables, layout paths, and devices. To avoid the issue, one of four QEC output clocks is selected by an on-chip MUX and measured. As a result, the selected output clock $CLK_{OUT,P}$ is monitored by an oscilloscope with a trigger clock having the same frequency as four phase clocks.

The time difference between the rising edge of the trigger clock and the rising edge of $CLK_{OUT,P}$ is measured with the oscilloscope as shown in Figure 6.23. For

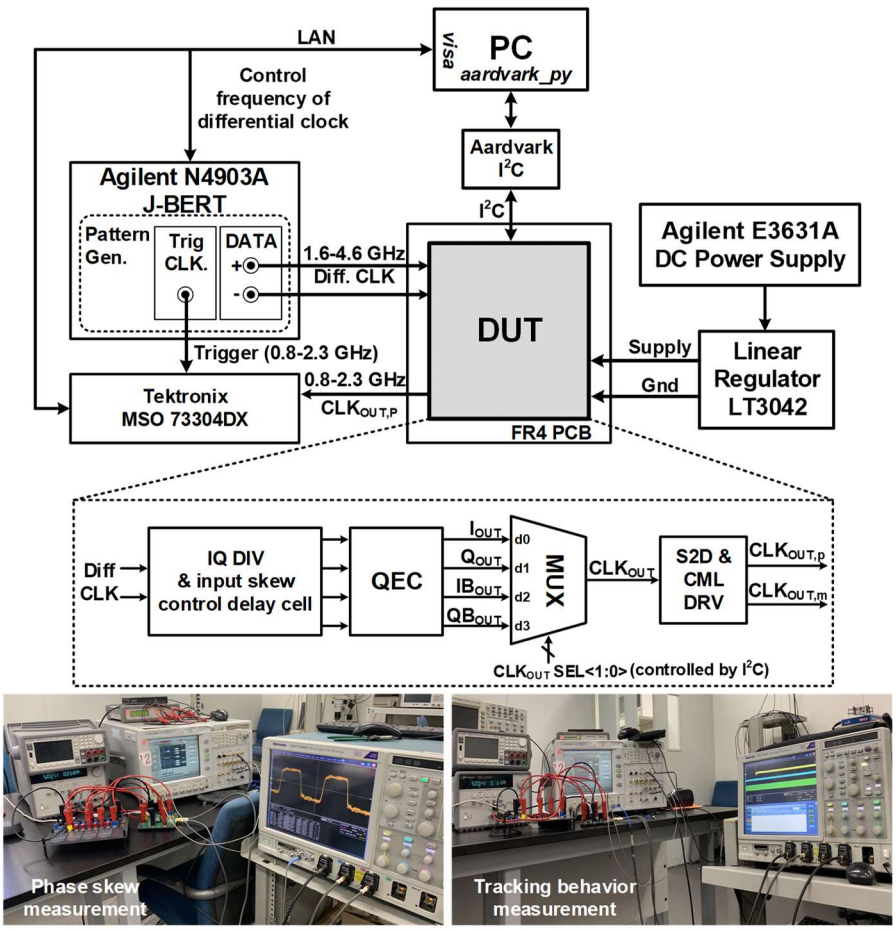


Fig. 6.22 Measurement setup of the proposed QEC.

example, Δt_I indicates the time difference between the trigger clock and I_{OUT}. Using information of the time differences, phase difference between quadrature clocks can be gotten.

The maximum phase error is calculated, which means the largest value of the phase error between adjacent clocks. The maximum phase error measured without and with the QEC in four different input phase error cases is shown in Figure 6.24. With the QEC, the maximum output phase error from 0.8-to-2.3 GHz is lower than 2.18° when the range of the input phase error is 101.6 ps. In Figure 6.25, the measured root-mean-square (RMS) jitter of the 2.3 GHz output clock without and with the QEC is 2.28 ps_{RMS} and 2.34 ps_{RMS}, respectively. In the case without the QEC, all the codes of DCDLs in the main clock path are fixed at the minimum to estimate the RMS jitter increased by the QEC due to the increased delay. The QEC contributes the amount of RMS jitter to 0.53 ps_{RMS} because it minimizes the increased delay in the main clock path.

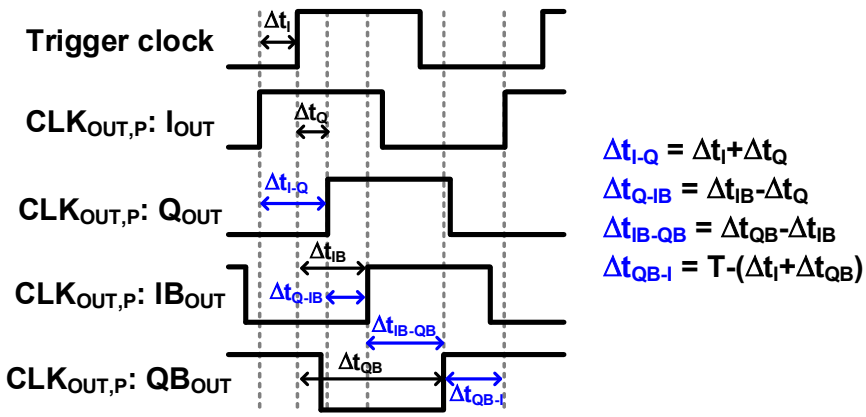
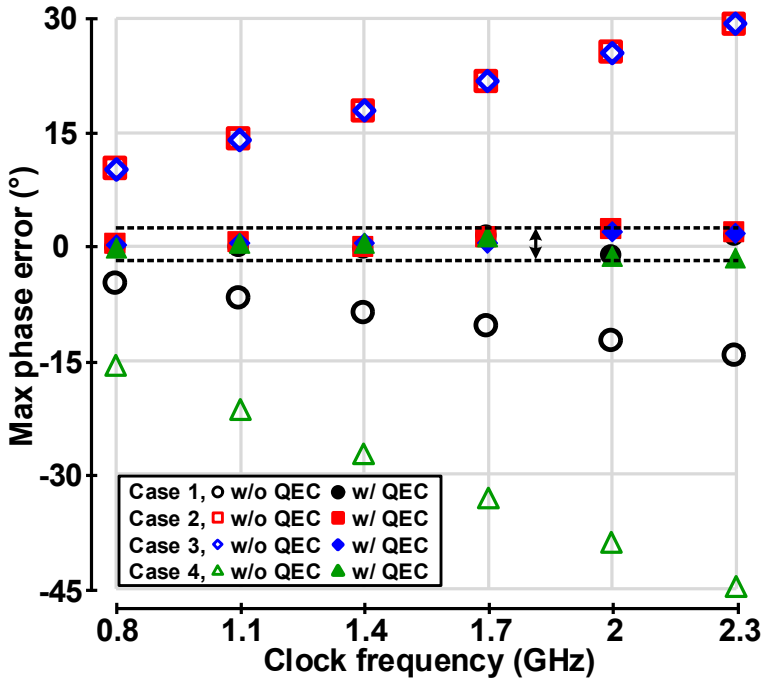


Fig. 6.23 Phase skew measurement method.

Figure 6.26 shows the tracking behavior of the proposed QEC. At 2.3 GHz, QB_{OUT} is selected when the transient measure is carried out. Thanks to the binary search, the locking time of the QEC is about 500 ns.

In Figure 6.27, the bar graph shows the power consumption when the calibration loop is turned on and off, respectively, after the calibration is complete. The power consumption of the calibration loop is reduced from 5.12 mW to 0.13 mW at 2.3



Input phase error	$\Delta t_{I\text{-}Q,err}$	$\Delta t_{Q\text{-}IB,err}$	$\Delta t_{IB\text{-}QB,err}$	$\Delta t_{QB\text{-}I,err}$
Case 1	8.87 ps	-17.4 ps	8.97 ps	-0.4 ps
Case 2	-29 ps	-16.2 ps	9.95 ps	35.2 ps
Case 3	-28.7 ps	20.3 ps	-27 ps	35.4 ps
Case 4	47.4 ps	-54.2 ps	45.8 ps	-39 ps

Fig. 6.24 Measured maximum phase error and 4 cases of input phase error.

GHz when excluding the power consumption of the main clock paths which are always turned on.

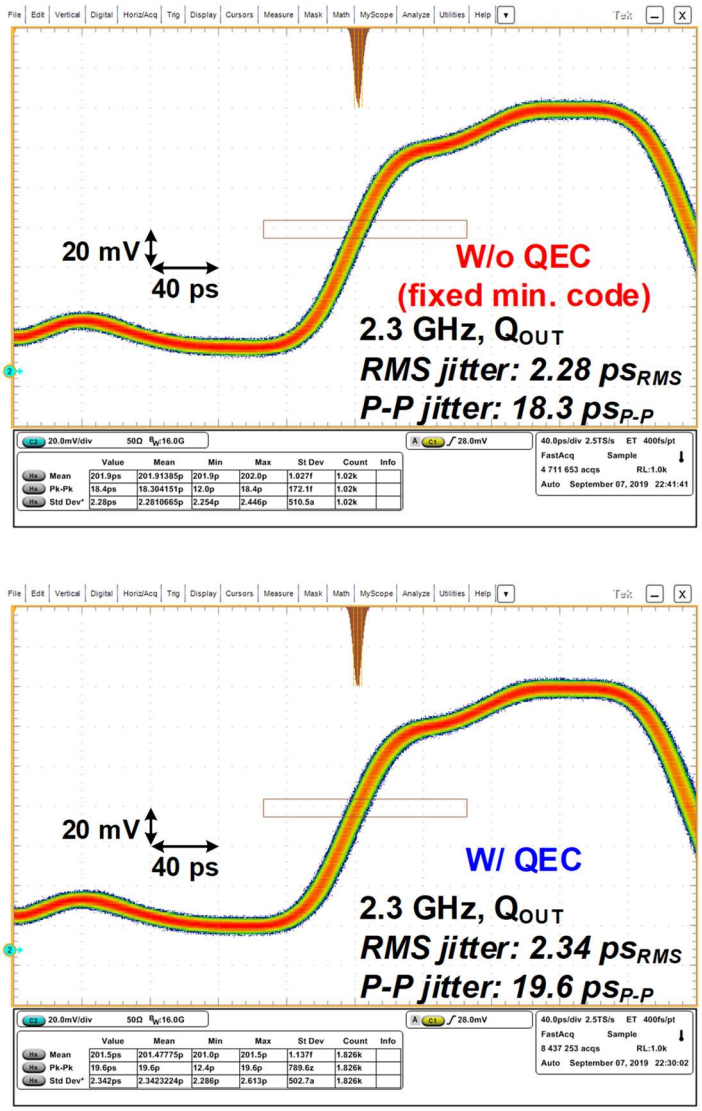


Fig. 6.25 Measured jitter histogram of 2.3 GHz Q_{OUT} without and with QEC.

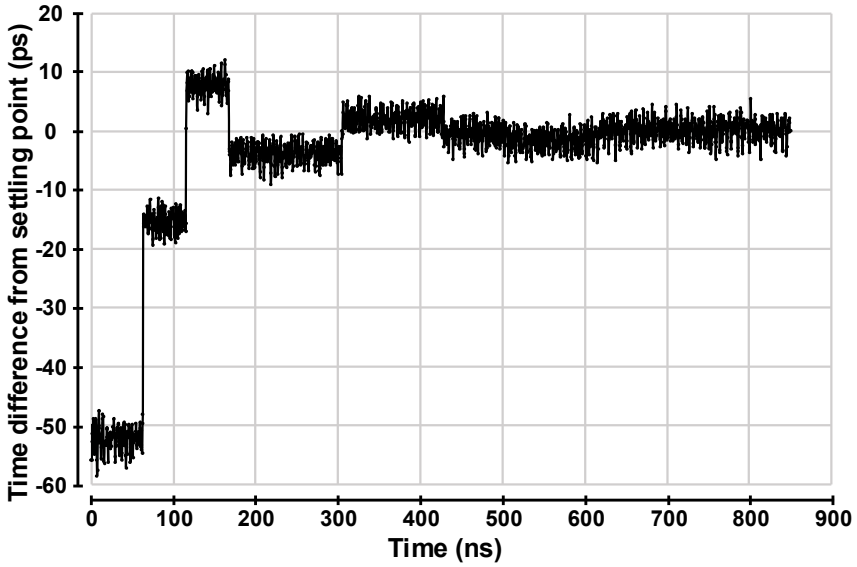


Fig. 6.26 Measured tracking behavior of the proposed QEC.

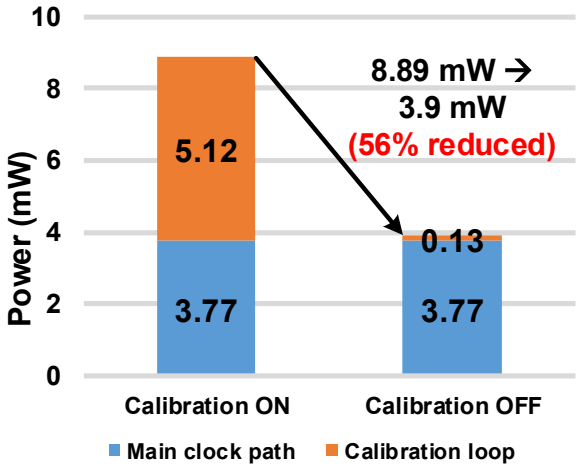


Fig. 6.27 Power consumption of the proposed QEC when the calibration loop is on and off.

Table 6.1 shows the performance summary of the proposed QEC and comparison with other state of the arts. Although the QEC consumes large power due to the large delay range of DCDLs for DRAM applications, the QEC achieves the lowest normalized RMS jitter contribution and has the best phase error correction ratio among other phase correctors.

Table 6.1 Performance comparison

	JSSC'09 [8]	ISSCC'09 [7]	TCAS2'17 [10]	TVLSI'19 [9]	This work
Technology	90 nm	45 nm	65 nm	55 nm	40 nm
VDD (V)	1	0.9	1	1.2	1.1
Architecture	Analog DLL	Open-loop	Digital DLL	Relaxation oscillator based PD	Digital DLL
Clock frequency (GHz)	8-10	0.8-5	1.25	1-3	0.8-2.3
Phase	5	4	4	4	4
RMS jitter before/after correction (ps_{RMS})	0.257/0.293 (10 GHz)	0.81/0.94 (5 GHz)	1.8/2.5	1.85/2.14 (3 GHz)	2.28/2.34 (2.3 GHz)
RMS jitter contribution (ps_{RMS}) (Normalized RMS jitter contribution)*	0.14 (1.4)	0.48 (2.4)	1.73 (2.16)	1.08 (3.24)	**0.53 (1.22)**
P-P jitter before/after correction (ps_{P-P})	N/A /2.04	7.1/8	12.7/15	15.75/19.75	18.3/19.6 (2.3 GHz)
Correctable phase error range	26°** @ 9.5 GHz	N/A	8.7°	< 24°** @ 3 GHz	**84.1° @ 2.3 GHz**
Max. phase error after correction	4.79° @ 9.5 GHz	< 5° (0.8-5 GHz)	0.48°	0.7° @ 1 GHz 1.1° @ 3 GHz	< 2.18° (0.8-2.3 GHz) 1.77° @ 2.3 GHz
Phase error correction ratio***	5.43 @ 9.5 GHz	N/A	18.13	21.82 @ 3 GHz	**47.53 @ 2.3 GHz**
Power (mW)	15	5.4 @ 5 GHz	2.27	2.08 @ 3 GHz	8.89 (cal on) 3.9 (cal off) @ 2.3 GHz
Calibration loop power (mW)	N/A	5.4	N/A	1.41****	5.12****
Area (mm^2)	0.03	0.0035	0.01	0.003	0.0428
FOM*****	0.39	N/A	0.22	0.10	**0.10 (cal on)** **0.04 (cal off)**

*RMS jitter contribution (ps_{RMS})·clock frequency (GHz)

**Estimated from paper

***Phase error correction ratio = correctable phase error range/max. phase error after correction

****Without DCDL power of clock paths

*****FOM = Normalized RMS jitter contribution·power (mW)/(clock frequency (GHz)·phase error correction ratio)

Chapter 7

Conclusion

In this thesis, the forwarded-clock (FC) receiver with DLL-based self-tracking loop, the open-loop-based voltage drift compensation in clock distribution, and the quadrature error corrector are proposed. The proposed FC receiver adopts a cascaded DLL architecture to support the DRAM interface. The proposed receiver compensates for the VT drift by fixing tDQS2DQ using a DLL. Therefore, it does not require re-training in the memory controller. Since it utilizes the write training, it does not need DQ transitions. Besides, it does not increase the capacitance at DQ pins because monitoring DQ is not necessary for VT drift compensation. The proposed FC receiver is implemented in a 65-nm CMOS process and 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.

An open-loop-based method is proposed for voltage drift compensation in the clock distribution. The open-loop-based compensation scheme has a simple architec-

ture and minimizes increased power and occupied area due to the compensation scheme. The bias voltages generated by the proposed bias generator control inverters and CML-to-CMOS converters in the CMOS path to compensate for voltage drift. The clock tree applying the proposed method is fabricated in 40 nm CMOS process. When the supply voltage is 1.1 V and the supply modulation is 1 MHz 100 mV_{P-P} swing sinusoidal wave, the measured RMS jitter of the conventional scheme is 3.77 ps_{RMS}. Under the same condition, the measured RMS jitter of the proposed scheme is 1.61 ps_{RMS}. The power consumption of the proposed scheme is 11.02 mW at 6 GHz output clock.

The quadrature error corrector (QEC) with an operating frequency range of 0.8 to 2.3 GHz is proposed. It minimizes the added delays in the main clock path, thus minimizing the additional jitter caused by DCDLs for phase error correction. The amount of the contributed jitter by QEC is 0.53 ps_{RMS} at 2.3 GHz. Also, the use of the asynchronous calibration on-off scheme reduces the power consumption from 8.89 mW to 3.9 mW after calibration at 2.3 GHz. The proposed QEC is fabricated in 40 nm CMOS. The measured correctable phase error range achieves 101.6 ps and the remaining phase error after error correction is less than 2.18°.

Bibliography

[1] H. Park *et al*., "A 3-bit/2UI 27Gb/s PAM-3 single-ended transceiver using one-tap DFE for next-generation memory interface," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2019, pp. 382–383.

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

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

[4] S. Shin, H.-G. Ko, C.-H. Kye, S.-Y. Lee, J. Yun, D. Lee, H.-K. Jung, S. Kim, and D.-K. Jeong, "A 0.45 pJ/b, 6.4 Gb/s forwarded-clock receiver with DLL-based self-tracking loop for unmatched memory interfaces," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 67, no. 10, pp. 1814–1818, Oct. 2020.

[5] M. Mansuri and C.-K. K. Yang, "A low-power adaptive bandwidth PLL and clock buffer with supply-noise compensation," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1804–1812, Nov. 2003.

[6] M.-C. Chou, "Delay circuit with constant time delay independent of temperature variations," U.S. Patent 7 932 764, Apr. 26, 2011.

[7] K.-H. Kim *et al*., "A 5.4mW 0.0035mm^2 0.48ps$_{rms}$-jitter 0.8-to-5GHz non-PLL/DLL all-digital phase generator/rotator in 45nm SOI CMOS," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2009, pp. 98–99.

[8] K.-J. Hsiao and T.-C. Lee, "An 8-GHz to 10-GHz distributed DLL for multiphase clock generation," *IEEE J. Solid-State Circuits*, vol. 44, no. 9, pp. 2478–2487, Sep. 2009.

[9] J.-H. Chae, H. Ko, J. Park, and S. Kim, "A quadrature clock corrector for

DRAM interfaces, with a duty-cycle and quadrature phase detector based on a relaxation oscillator," *IEEE Trans. Very Large Scale Integration Syst*., vol. 27, no. 4, pp. 978–982, Apr. 2019.

[10] Y. Kim, K. Song, D. Kim, and S. Cho, "A 2.3-mW 0.01-mm^2 1.25-GHz quadrature signal corrector with 1.1-ps error for mobile DRAM interface in 65-nm CMOS," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 64, no. 4, pp. 397–401, Apr. 2017.

[11] P. Kinget, R. Melville, D. Long, and V. Gopinathan, "An injection-locking scheme for precision quadrature generation," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 845–851, July 2002.

[12] S. Shin, H.-G. Ko, S. Jang, D. Kim, and D.-K. Jeong, "A 0.8-to-2.3GHz quadrature error corrector with correctable error range of 101.6ps using minimum total delay tracking and asynchronous calibration on-off scheme for DRAM interface," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2020, pp. 340–341.

[13] C. Kim, H.-W. Lee, and J. Song, *High-Bandwidth Memory Interface*. Heidelberg, Germany: Springer, 2014.

[14] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Singapore, Singapore: McGraw-Hill, 2001.

[15] D. D. Caro, "Glitch-free NAND-based digitally controlled delay-lines," *IEEE Trans. Very Large Scale Integration Syst*., vol. 21, no. 1, pp. 55–66, Jan. 2013.

[16] Y. Lee and I.-C. Park, "Single-step glitch-free NAND-based digitally controlled delay lines using dual loops," *Electronics Letters*, vol. 50, no. 13, pp. 930–932, June 2014.

[17] J. S. Lee *et al*., "A wide range PLL for 64X speed CD-ROMs & 10X speed DVD-ROMs," *IEEE Trans. Consumer Electronics*, vol. 46, no. 3, pp. 487–493, Aug. 2000.

[18] M.-S. Hwang, J. Kim, and D.-K. Jeong, “Reduction of pump current mismatch in charge-pump PLL,” *Electronics Letters*, vol. 45, no. 3, pp. 135–136, Jan. 2009.

[19] S.-K. Lee, B. Kim, H.-J. Park, J.-Y. Sim, “A QDR-based 6-GB/s parallel transceiver with current-regulated voltage-mode output driver and byte CDR for memory interface,” *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 60, no. 2, pp. 91–95, Feb. 2013.

[20] *LPDDR4 Specification (JESD209-4)*, JEDEC Standard 209-4, Aug. 2014.

[21] 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*, June 2019, pp. C94–C95.

[22] Y.-S. Kim *et al*., “An 8GB/s quad-skew-cancelling parallel transceiver in 90nm CMOS for high-speed DRAM interface,” *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2012, pp. 136–137.

[23] 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 Integrated Circuits Conf*., Sep. 2013, pp. 1–4.

[24] 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*, June 2014, pp. 1–2.

[25] 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 and Syst. II: Exp. Briefs*, vol. 63, no. 3, pp. 264–268, Mar. 2016.

[26] W. Bae *et al*., “A 0.36 pJ/bit, 0.025 mm^2, 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 and Syst. I: Reg. Papers*, vol. 63, no. 9, pp. 1393–1403, Sep. 2016.

[27] B. Casper and F. O'Mahony, "Clocking analysis, implementation and measurement techniques for high-speed data links-a tutorial," *IEEE Trans. Circuits and Syst. I: Reg. Papers*, vol. 56, no. 1, pp. 17–39, Jan. 2009.

[28] Y.-C. Huang, C.-F. Liang, H.-S. Huang, P.-Y. Wang, "A 2.4GHz ADPLL with digital-regulated supply-noise-insensitive and temperature-self-compensated ring DCO," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2014, pp. 270–271.

[29] C. Menolfi *et al*., "A 16Gb/s source-series terminated transmitter in 65nm CMOS SOI," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2007, pp. 446–447.

[30] U. Singh and M. Green, "Dynamics of high-frequency CMOS dividers," *in Proc. IEEE Int. Symp. Circuits and Syst*., May 2002, pp. 421–424.

[31] T. Wu, K. Mayaram, and U.-K. Moon, "An on-chip calibration technique for reducing supply voltage sensitivity in ring oscillators," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 775–783, Apr. 2007.

[32] H.-W. Lee *et al*., "A 1.0-ns/1.0-V delay-locked loop with racing mode and countered CAS latency controller for DRAM interfaces," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1436–1447, June 2012.

[33] S. Kumaki *et al*., "A 0.5V 6-bit scalable phase interpolator," *in Proc. IEEE Asia Pacific Conf. Circuits and Syst*., Dec. 2010, pp. 1019–1022.

[34] N. D. Dalt, "A design-oriented study of the nonlinear dynamics of digital bang-bang PLLs," *IEEE Trans. Circuits and Syst. I: Reg. Papers*, vol. 52, no. 1, pp. 21–31, Jan. 2005.

초 록

본 논문에서는 동적 랜덤 액세스 메모리 (DRAM)의 속도가 증가함에 따라 클록 패스에서 발생할 수 있는 문제에 대처하기 위한 세 가지 회로들을 제안하였다. 제안한 회로들 중 두 방식들은 지연동기루프 (delay-locked loop) 방식을 사용하였고 나머지 한 방식은 면적과 전력 소모를 줄이기 위해 오픈 루프 방식을 사용하였다. DRAM 의 비정합 수신기 구조에서 데이터 패스와 클록 패스 간의 지연 불일치로 인해 전압 및 온도 변화에 따라 셋업 타임 및 홀드 타임이 줄어드는 문제를 해결하기 위해 지연동기루프를 사용하였다. 제안한 지연동기루프 회로는 DRAM 환경에서 동작하도록 두 개의 지연동기루프로 나누었다. 또한 초기 쓰기 훈련을 통해 데이터와 클록을 타이밍 마진 관점에서 최적의 위치에 둘 수 있다. 따라서 제안하는 방식은 데이터 천이 정보가 필요하지 않다. 65-nm CMOS 공정을 이용하여 만들어진 칩은 6.4 Gb/s 에서 0.45 pJ/bit 의 에너지 효율을 가진다. 또한 1 V 에서 쓰기 훈련 및 지연동기루프를 고정시키고 0.94 V 에서 1.06 V 까지 공급 전압이 바뀌었을 때 타이밍 마진은 0.31 UI 보다 큰 값을 유지하였다.

다음으로 제안하는 회로는 클록 분포 트리에서 전압 변화로 인해 클록 패스의 지연이 달라지는 것을 앞서 제시한 방식과 달리 오픈 루프 방식으로 보상하였다. 기존 클록 패스의 인버터와 CML-to-CMOS 변환기의 구조를 변경하여 바이어스 생성 회로에서 생성한 공급 전압에 따라 바뀌는 바이어스 전압을 가지고 지연을 조절할 수 있게 하였다. 40-nm CMOS 공정을 이용하여 만들어진 칩의 6 GHz 클록에서의 전력 소모는 11.02 mW 로 측정되었다. 1.1 V 중심으로 1 MHz, 100 mV 피크 투 피크를 가지는 사인파

성분으로 공급 전압을 변조하였을 때 제안한 방식에서의 지터는 기존 방식의 3.77 ps_{RMS} 에서 1.61 ps_{RMS} 로 줄어들었다.

DRAM 의 송신기 구조에서 다중 위상 클록 간의 위상 오차는 송신된 데이터의 데이터 유효 창을 감소시킨다. 이를 해결하기 위해 지연동기루프를 도입하게 되면 증가된 지연으로 인해 위상이 교정된 클록에서 지터가 증가한다. 본 논문에서는 증가된 지터를 최소화하기 위해 위상 교정으로 인해 증가된 지연을 최소화하는 위상 교정 회로를 제시하였다. 또한 유휴 상태에서 전력 소모를 줄이기 위해 위상 오차를 교정하는 회로를 입력 클록과 비동기식으로 끌 수 있는 방법 또한 제안하였다. 40-nm CMOS 공정을 이용하여 만들어진 칩의 위상 교정 범위는 101.6 ps 이고 0.8 GHz 부터 2.3 GHz 까지의 동작 주파수 범위에서 위상 교정기의 출력 클록의 위상 오차는 2.18°보다 작다. 제안하는 위상 교정 회로로 인해 추가된 지터는 2.3 GHz 에서 0.53 ps_{RMS} 이고 교정 회로를 껐을 때 전력 소모는 교정 회로가 켜졌을 때인 8.89 mW 에서 3.39 mW 로 줄어들었다.

주요어 : 클록 트리, 지연동기루프, 위상 오차, 위상 오차 교정기, 비정합 수신기, 온도 변화, 전압 변화.

학 번 : 2016-20926