

## Research Article

# 480 MHz 10-tap Clock Generator Using Edge-Combiner DLL for USB 2.0 Applications

Takashi Kawamoto,<sup>1</sup> Kazuhiro Ueda,<sup>2</sup> and Takayuki Noto<sup>3</sup>

<sup>1</sup>Hitachi Central Research Laboratory, 1-280, Higashi-Koigakubo Kokubunji-shi, Tokyo 185-8601, Japan

<sup>2</sup>Technology Development Unit, Mixed Signal Core Development Division, Converter Development Department, Renesas Electronics Corporation, 4-1 Mizuhara, Itami-shi, Hyogo 664-0005, Tokyo, Japan

<sup>3</sup>SoC Business Unit, 1st Industry & Network Business Division, Optical Disc Solutions Department, Renesas Electronics Corporation, 5-20-1 Josuichon-cho, Kodaira-shi, Tokyo 187-8588, Japan

Correspondence should be addressed to Takashi Kawamoto, takashi.kawamoto.hv@hitachi.com

Received 28 June 2011; Accepted 8 September 2011

Academic Editor: Woogeun Rhee

Copyright © 2012 Takashi Kawamoto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A clock generator with an edge-combiner DLL (ECDLL) has been developed for USB 2.0 applications. The clock generator generates 480 MHz 10-tap output signals from a 12 MHz reference signal and consists of three DLLs to shrink the design area so that it is smaller than a conventional one based on a PLL. Each DLL is applied to our proposed shot pulse reset technique to prevent from a harmonic lock and is applied to a voltage-controlled delay line (VCDL) with a trimming function to operate against any process voltage temperature (PVT) variations. A 90 nm CMOS process was used to fabricate our proposed clock generator. The 480 MHz 10-tap output signals satisfy the USB 2.0 specifications. A power consumption is less than 1.3 mW and a locking time is less than 3.5  $\mu$ s, which are far less than a conventional one, 10.0  $\mu$ s. The design area is  $200 \times 225 \mu$ m, which is half that of the conventional one.

## 1. Introduction

The clock generator may be one of the largest blocks of the physical layer (PHY) in wireline communications, because it usually consists of a phase-locked loop (PLL) that has large capacitors for use as a lowpass filter. There are many reports proposing the shrinkage of the design area of PLLs. A capacitance-multiplication technique was reported to shrink the capacitance of the loop filter [1, 2]. However, the shrink ratio of a capacitor may be less than five when taking the leakage current of the capacitor and PVT variation into consideration. Thus, the total design area cannot be drastically reduced. An all digital PLL (ADPLL) technique has also been reported [3]. However, an issue with the accuracy operation against the PVT variation still remains. Therefore, a new approach is desirable for essentially reducing the design area.

The DLL has several advantages over the PLL. First, it can be designed to be smaller than the PLL. While the PLL is a higher-order system, the DLL is a first-order system and is

always stable. Thus, the DLL needs small capacitors to keep the DLL loop stable while the PLL needs large capacitors to design a stable lowpass filter. Second, the DLL can achieve a shorter locking time than the PLL. Third, the DLL consumes less power than the PLL. The PLL has the VCO and a divider that consumes a large amount of power in order to reduce the jitter. However, the DLL has several disadvantages over the PLL when used as the clock generator. First, the DLL cannot generate faster clock signals than the PLL. Second, the DLL has a locking range limitation while the PLL does not. This means that the DLL cannot achieve a fractional multiplication ratio while the PLL can achieve a fractional- $N$  PLL.

An edge-combiner DLL (ECDLL) has been reported as an alternate high-speed clock generator because the ECDLL is based on the DLL and can multiply the reference frequency [4–8]. The ECDLL has a potential for use as the clock generator although it has barely been used in this capacity because the ECDLL has several challenges that need to be

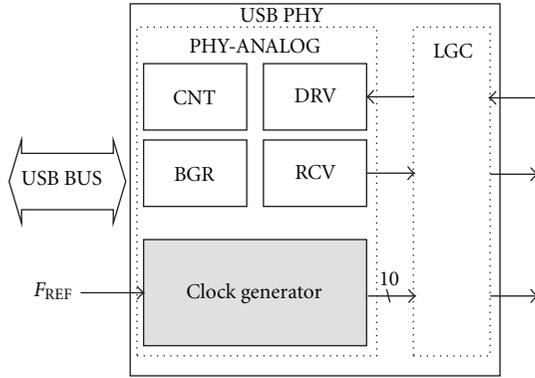


FIGURE 1: Block diagram of USB 2.0 PHY.

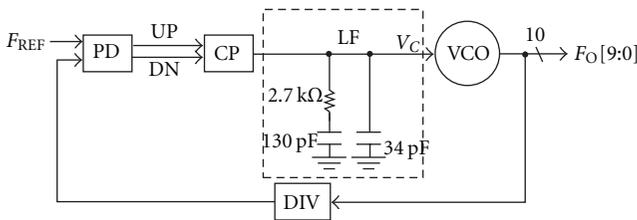


FIGURE 2: Conventional clock generator based on PLL.

overcome. The first is an operation against PVT variation. The second is the output signal frequency limitation.

From the viewpoint of the frequency limitation, the operation frequency of the DLL has been increasing in recent CMOS process. The DLL can operate at less than 1 GHz in a submicron CMOS process. Thus, the DLL might be able to be use as the clock generator for wireline communications whose operation frequency is less than about 1 GHz. USB 2.0 is the most popular wireline communication in the world and operates at 480 MHz. USB 2.0 PHY needs a small design area and a low power consumption level for use in portable devices. Therefore, USB 2.0 may be one of the most suitable applications for the clock generator with the ECDLL.

In order to apply the ECDLL for USB 2.0, we propose techniques that overcome the above-mentioned challenges. The first is a shot pulse generator to prevent from a harmonic lock. The third is the VCDL trimming function to operate against PVT variation.

In this paper, we propose a clock generator applied to an ECDLL for USB 2.0 PHY to shrink the design area [1]. The organization of this paper is as follows. Section 2 describes the overall structure of the proposed clock generator architecture. Section 3 describes the ECDLL and DLL in detail. Section 4 presents the evaluation results of our measurements, and Section 5 concludes with a short summary of the key points.

## 2. Overall Clock Generator Architecture

Figure 1 shows a block diagram of a USB 2.0 PHY. The PHY consists of a clock generator, a band-gap reference (BGR), a controller (CNT), a driver (DRV), a receiver (RCV), and

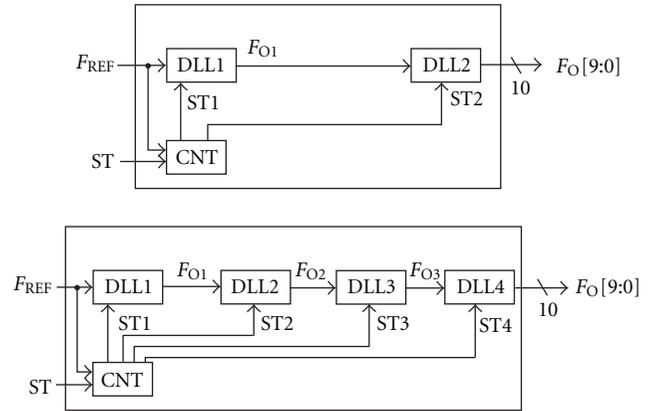


FIGURE 3: Block diagrams of candidate clock generator based on DLLs. Upper block diagram is the structure that consists of one ECDLL and one DLL. The ECDLL has a multiplication ratio of 40. Bottom block diagram is the structure that consists of three ECDLL and one DLL. The DLL1, DLL2, and DLL3 have multiplication ratio of two, four, and five, respectively.

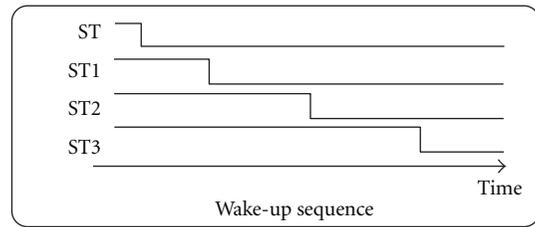
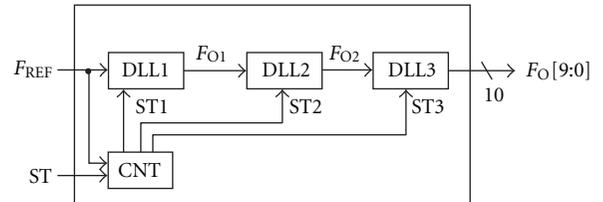


FIGURE 4: Proposed clock generator consisting of three DLLs and wake-up sequence. The DLL1 and DLL2 have multiplication ratio of two and eight, respectively.

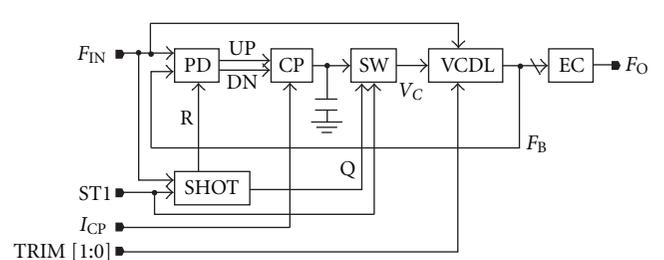


FIGURE 5: Block diagram of DLL1 and DLL2. In DLL1,  $F_{IN}$  and  $F_O$  are 12 MHz and 60 MHz, respectively. The VCDL generates 10-tap 12 MHz output signals. The capacitor is 10 pF. In DLL2,  $F_{IN}$  and  $F_O$  are 60 MHz and 480 MHz, respectively. The VCDL generates 16-tap 60 MHz output signals. The capacitor is 0.5 pF.

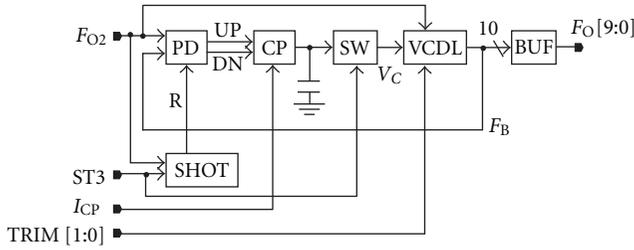


FIGURE 6: Block diagram of DLL.  $F_{O2}$  and  $F_{O[9:0]}$  are 480 MHz and 10-tap 480 MHz, respectively. The capacitor is 1 pF.

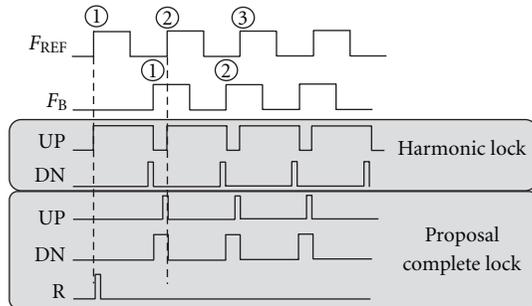


FIGURE 7: Explanation of harmonic lock and proposed shot pulse reset operation.

a logic block (LGC). Figure 2 shows a block diagram of a conventional clock generator based on the PLL with a ring oscillator. The reference signal ( $F_{REF}$ ) is 12 MHz. The output signals ( $F_{O[9:0]}$ ) are 10-tap 480 MHz. The design area is  $200 \times 550 \mu\text{m}^2$ . The loop bandwidth is designed at 1.6 MHz using a second-order lowpass filter that consists of 130 pF and 34 pF capacitances and a 2.7 k $\Omega$  resistance. Thus, the lowpass filter occupies a large portion of the design area. Therefore, we proposed the ECDLL as a clock generator to shrink the design area.

In this clock generator, there are three candidates, which are one ECDLL and one DLL, two ECDLL and one DLL, and three ECDLL and one DLL, as shown in Figures 3 and 4. The one ECDLL and one DLL structure consists of the DLL1 of the ECDLL that has the multiplication ratio of 40 and the DLL2 of the DLL that generates the 10-tap 480 MHz signals, as shown in Figure 3. The two ECDLL and one DLL structure consists of the DLL1 of the ECDLL that has the multiplication ratio of five, the DLL2 of the ECDLL that has the multiplication ratio of eight, and the DLL3 of the DLL that has the same manner of the above DLL, as shown in Figure 4. The three ECDLL and one DLL structure consists of the DLL1 of the ECDLL that has the multiplication ratio of two, the DLL2 of the ECDLL that has the multiplication ratio of four, the DLL3 of the ECDLL that has the multiplication ratio of five, and the DLL4 of the DLL that has the same manner of the above DLL, as shown in Figure 3.

First, it is reasonable that the ECDLL has the multiplication ratio of less than 10, according to the design area and operation against from PVT variation. As the multiplication ratio is larger, the number of the VCDL stage is larger. It

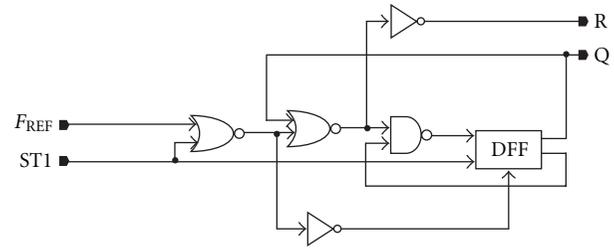


FIGURE 8: Block diagram of shot pulse generator.

causes large design area. The one ECDLL and one DLL structure becomes two times as large as the two ECDLL and one DLL structure because the ECDLL of the multiplication ratio of 40 is large. The two ECDLL and one DLL structure is almost same design area as the three ECDLL and one DLL structure. And then, the previous DLL operates more slowly than the latter one. Thus, the delay cell size in the previous DLL may be smaller than that in the latter one. Therefore, to shrink the design area, the previous DLL might have smaller multiplication ratio than the latter one.

Second, the number of the cascade DLL block should be as low as possible because the operation of the whole clock generator could be stable and the settling period could be short. In our proposed clock generator, the standby sequence is necessary because the DLL may fall into the unlock state if the DLL starts to operate before the previous DLL completes the lock, as shown in Figure 4.

Finally, the two ECDLL and one DLL structure is proposed, considering above concern. The DLL1 and DLL2 have a multiplication ratio of five and eight, respectively.

The counter (CNT) generates the standby signals of each block ( $ST1$ ,  $ST2$ , and  $ST3$ ) using the standby signal of the clock generator ( $ST$ ) to create a standby sequence for each DLL, as shown in Figure 4. If DLL2 starts the lock operation before DLL1 completes the lock, DLL2 might fall into the unlock state. Thus, the CNT controls the sequential wake-up operation by generating  $ST1$ ,  $ST2$ , and  $ST3$ , as shown in Figure 4.

### 3. ECDLL and DLL in Detail

**3.1. Shot Pulse Reset Technique.** Figures 5 and 6 show a block diagram of our proposed ECDLL applied to DLL1 and DLL2 and DLL applied to DLL3. They consist of a phase detector (PD), a charge-pump (CP), a switch (SW), a capacitor, a voltage-controlled delay line (VCDL), and shot pulse generator (SHOT). And then the ECDLL has an edge-combiner (EC) and the DLL has the output buffer (BUF). The PD makes a comparison between the phase of the  $F_{REF}$  and a phase of the feedback clock ( $F_B$ ) and generates result signals (UP/DN). The CP charges and discharges the capacitor. The VCDL generates the output signals from the  $F_{REF}$ . The delay time from  $F_{REF}$  to  $F_B$  is controlled by a controlled voltage ( $V_C$ ). In Figure 6, the EC generates the output signal ( $F_{O1}$ ) from VCDL output signals.

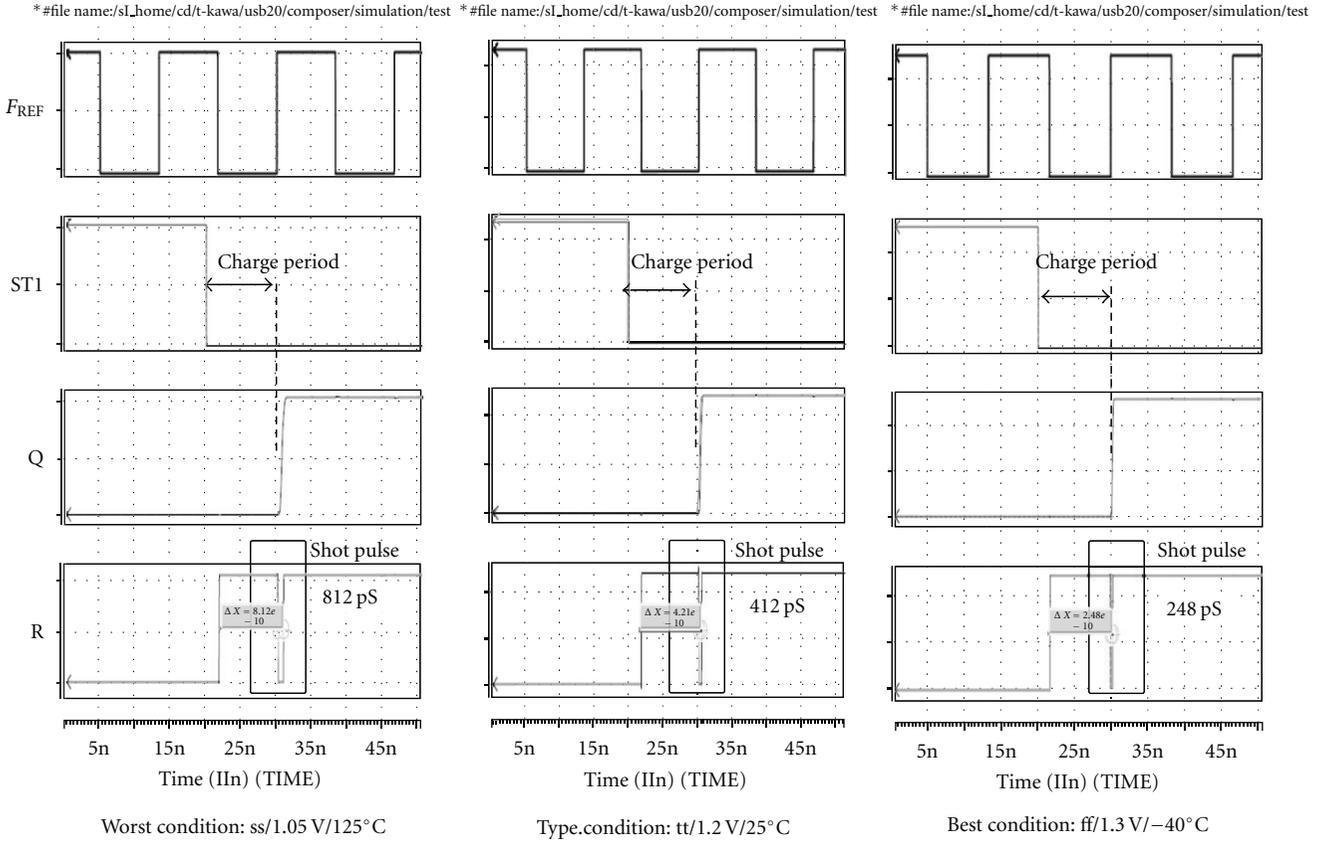


FIGURE 9: Simulation results from shot pulse generator.

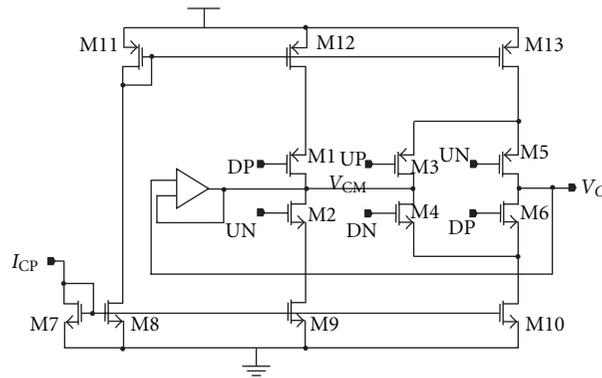


FIGURE 10: Circuit diagram of the CP.

Our ECDLL and DLL have an issue of a harmonic lock. We propose shot pulse reset technique by using the shot pulse generator to resolve this issue.

Figure 7 shows an explanation of the proposed shot pulse reset technique. The PD compares the rise edge of the  $F_{REF}$  and the rise edge of the  $F_B$  and generates the UP and DN signals. When the second rise edge of the  $F_{REF}$  comes after the first rise edge of the  $F_B$ , a harmonic lock occurs that the PD compares between first rise edge of the  $F_{REF}$  and the first rise edge of the  $F_B$ . It is an actual operation that the PD compares between the second rise edge of the  $F_{REF}$  and the first rise edge

of the  $F_B$ . A shot pulse reset shown in Figure 7 is proposed to prevent this malfunction from happening. The operation of the PD is reset by R when the PD compares the first rise edge of the  $F_{REF}$  with the first rise edge of the  $F_B$ . Thus, the PD can compare the second rise edge of the  $F_{REF}$  with the first rise edge of the  $F_B$ . Even though the PD is reset by R, the harmonic lock occurred and the PD compared the third rise edge of the  $F_{REF}$  with the first rise edge of the  $F_B$ , when the first rise edge of the  $F_B$  comes after the third rise edge of the  $F_{REF}$ . To prevent this malfunction from happening, the capacitor is charged by the SW and the voltage level of the

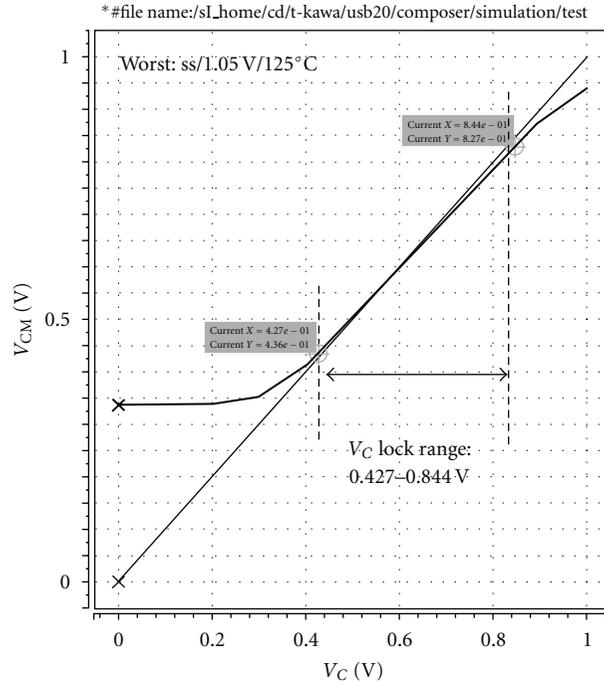


FIGURE 11: Simulation results from CP.

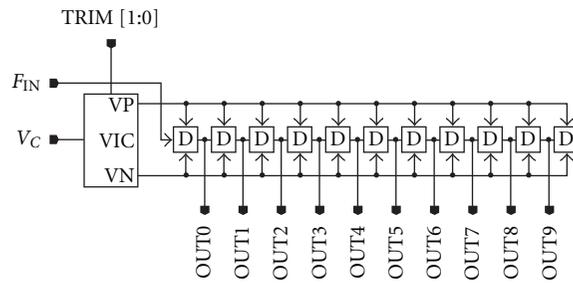


FIGURE 12: Block diagram of VCDL in DLL1 and DLL3.

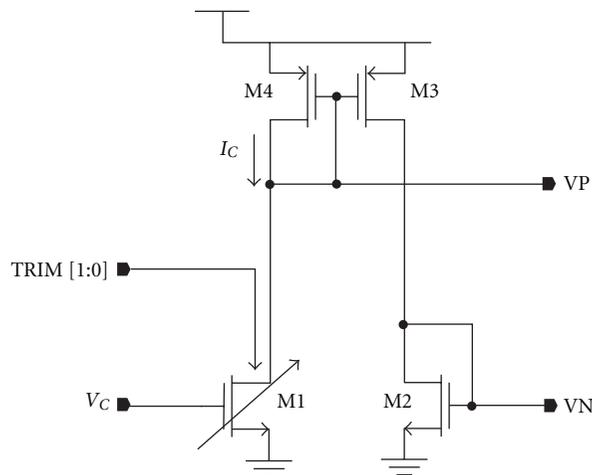


FIGURE 13: Circuit diagram of VIC.

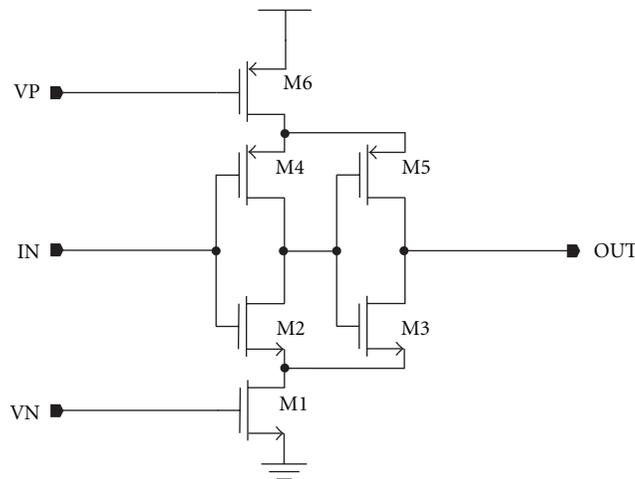


FIGURE 14: Circuit diagram of delay cell.

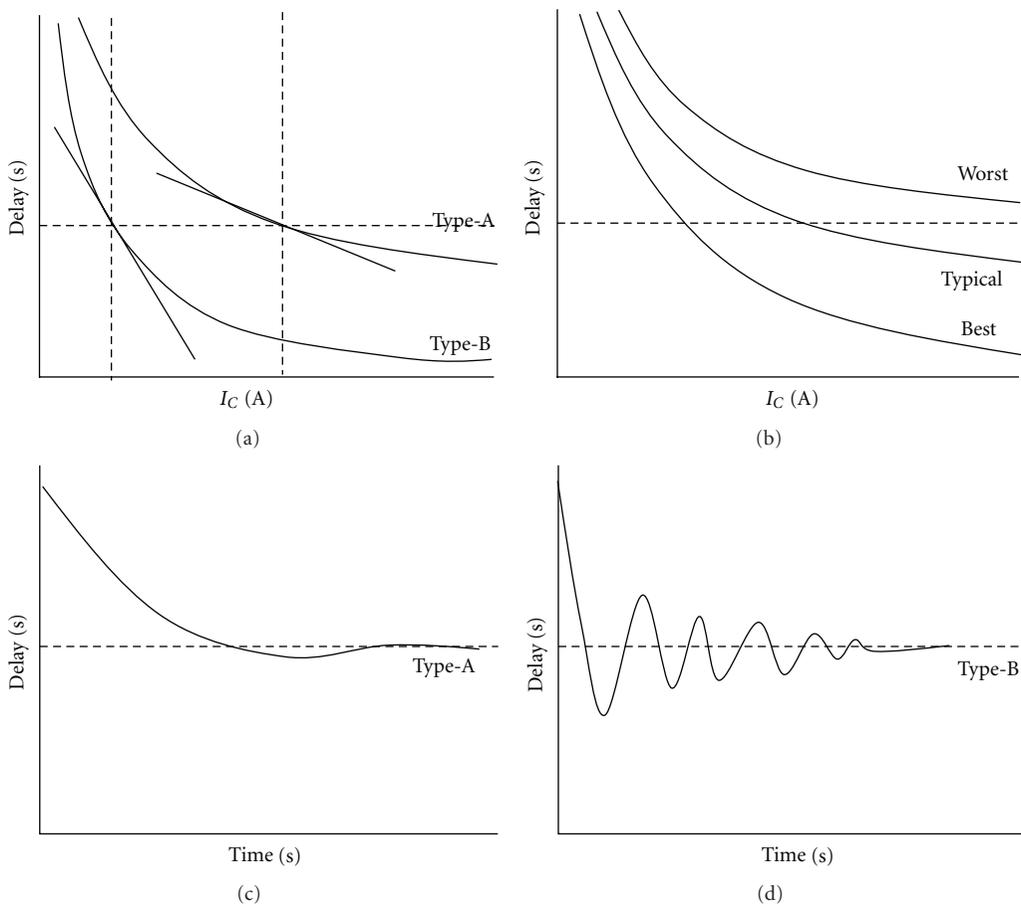


FIGURE 15: Explanation of the VCDL and DLL operation. (a) The explanation of the VCDL delay-current characteristics. The type-A VCDL has the smooth sensitivity and the type-B has the steep sensitivity. (b) The explanation of the influence of the VCDL delay-current characteristics due to the PVT variation. (c) The explanation of the DLL settling operation by using type-A VCDL. The settling operation is smooth because the sensitivity of the VCDL is not steep. (d) The explanation of the DLL settling operation by using type-B VCDL. The settling operation is not smooth because the sensitivity of the VCDL is steep.

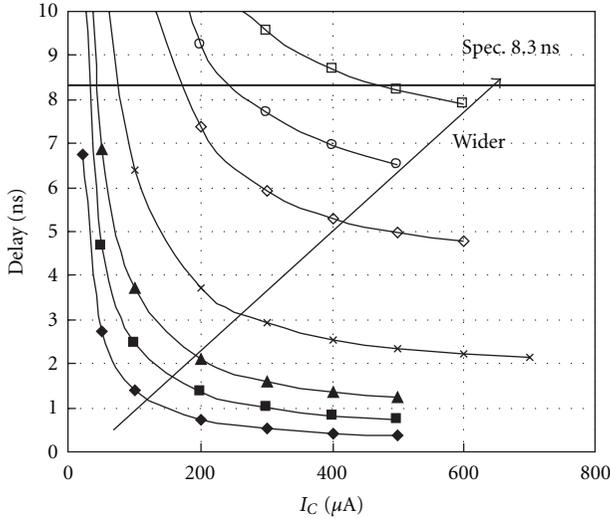


FIGURE 16: Postlayout simulation results of VCDL delay-current characteristics by using variable delay cell.

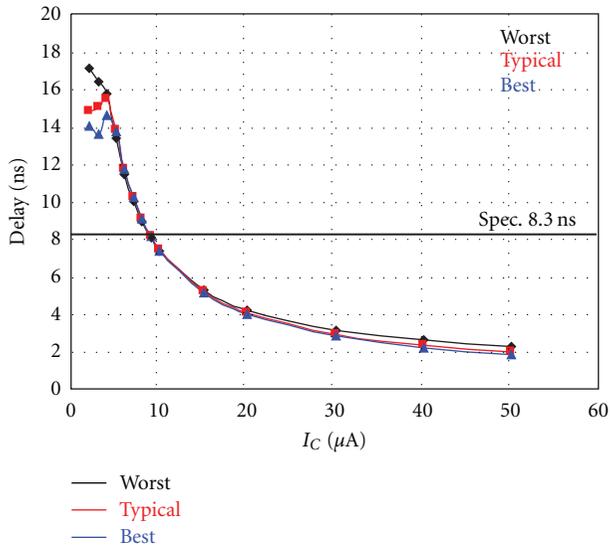


FIGURE 17: Postlayout simulation results of VCDL delay-current characteristics for DLL1.

$V_C$  is likely to be  $V_{DD}$ . This allows the first rise edge of the  $F_B$  to come before the second rise edge of the  $F_{REF}$  when the PD starts operation.

Figures 8 and 9 show a circuit diagram of the shot pulse generator (SHOT) and the simulation results from the SHOT. After the ST1 is set to low, the R operates the pulse reset and the Q is set to high at the rise edge of the  $F_{REF}$ . The pulse of the R resets the PD operation and the SW charges the capacitor during the period between the fall edge of the ST1 and the rise edge of the Q. After charging the capacitor, the  $V_C$  is almost the  $V_{DD}$ . Our ECDLL and DLL can be operated accurately because of the SHOT.

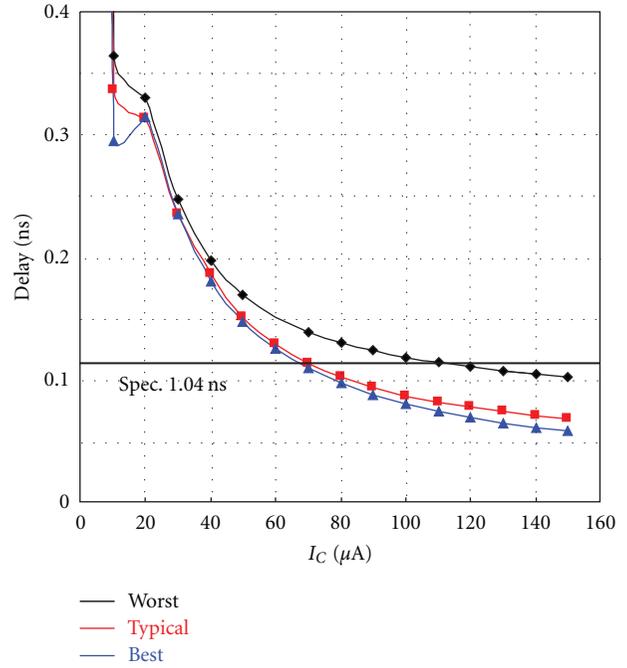


FIGURE 18: Postlayout simulation results of VCDL delay-current characteristics for DLL2.

**3.2. Charge Pump.** Figure 10 shows a circuit diagram of the CP. M5 and M6 are the switches that charge and discharge the capacitor. When the CP charges the current, the UP, UN, DP, and UN are high, low, low, and high, respectively. The charge current, which is the M13 drain current, passes through the switch M5 and charges the capacitor that is connected at the  $V_C$ , and the M12 drain current passes through the switches (the M1 and M4) and flows to M10. When the CP discharges the current, the UP, UN, DP, and UN are low, high, high, and low, respectively. The discharge current, which is the M10 drain current, passes through the switch at M6 and discharges the capacitor that is connected at the  $V_C$ , and the M13 drain current passes through the switches (the M3 and M2) and flows to M9. The Op-Amp is designed in the CP to structure the common-mode feedback. When the  $V_C$  is not equal to the voltage of the  $V_{CM}$ , the difference between the charge current and the discharge current is larger. This causes a constant phase error. Figure 11 shows the simulation results from the CP. The lock range, which is the range in which the  $V_C$  is equal to the voltage of the  $V_{CM}$ , is 0.427–0.884 V under the worst conditions (ss/1.05 V/125°C).

**3.3. VCDL.** Figure 12 shows a block diagram of the voltage-controlled delay line (VCDL) for DLL1 and DLL3. It consists of a voltage-current converter (VIC) and the delay chain consisting of eleven delay cells. The VCDL for DLL2 has the seventeen delay cells. Figures 13 and 14 show the circuit diagrams of the VIC and the delay cell. The VIC converts the control voltage ( $V_C$ ) to a control current ( $I_C$ ). The VCDL delays the  $F_{IN}$ , which is the amount of the delay controlled by the  $I_C$ . The trimming signal (TRIM[1:0]) controls the sensitivity of the current-voltage characteristics of the M1 by

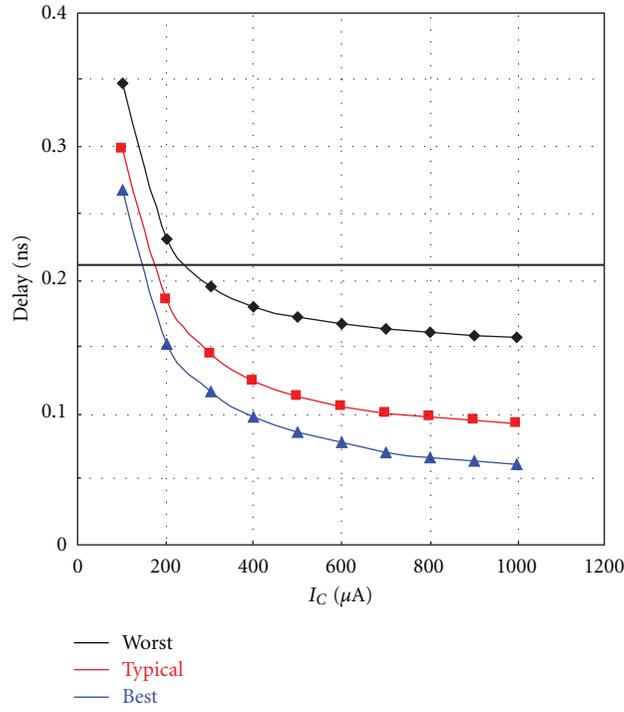


FIGURE 19: Postlayout simulation results of VCDL delay-current characteristics for DLL2.

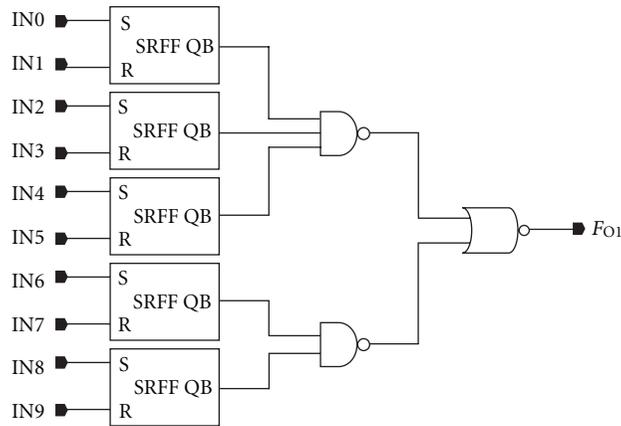


FIGURE 20: Block diagram of EC for DLL1.

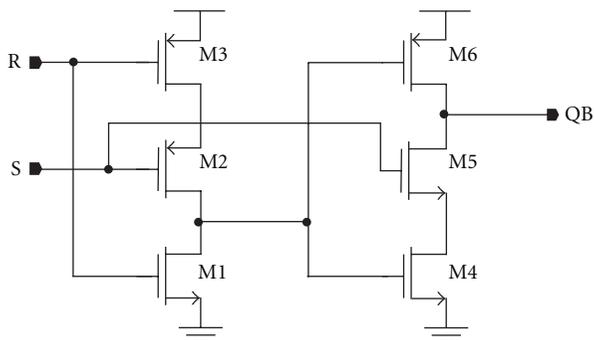


FIGURE 21: Circuit diagram of SRFF.

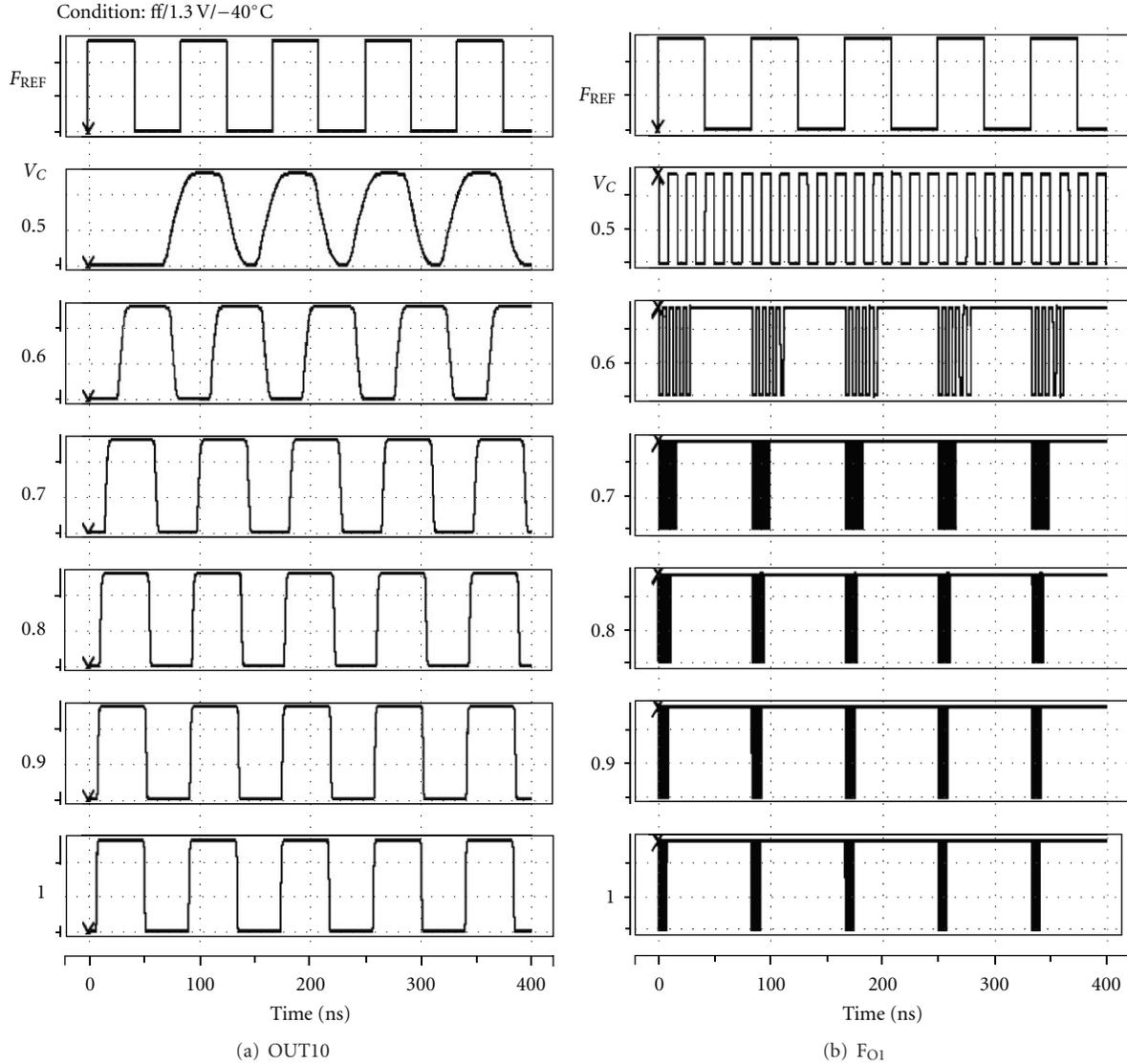


FIGURE 22: Postlayout simulation results from VCDL and EC applied to DLL1 for variable  $V_C$ . The simulation condition is ff/1.30 V/−40°C.

changing the gm of M1. The larger the TRIM[1:0] is, the larger the  $I_C$  is. The delay cell consists of the two inverter buffers (M4-M2 and M5-M6) and current sources (M6 and M1).

In the VCDL, the sensitivity of the VCDL is important for DLL operation. Figure 15 shows the explanation of the sensitivity of the VCDL and the DLL settling operation. If the sensitivity of the VCDL delay-current characteristics is larger at the lock point, the DLL settling operation may not be stable as shown in Figure 15(d). It is the reason that the overshoot is large because the magnitude of the delay change per one clock cycle is large. To prevent from this unstable state, the VCDL sensitivity is designed small by using large delay cell for VCDL, as shown in Figure 15(a). However, this design causes large power consumption and the malfunction may be caused in the worst condition if the sensitivity is designed too small, as shown in Figure 15(b).

The delay is mainly generated as the control current and input capacitor, which is gate capacitor of M4 and M2, and a parasitic capacitor between delay cells. If the buffer MOSs (M4-M2 and M5-M3) are designed small, the necessary delay is obtained by small current. However, this causes large sensitivity. Thus, the buffer MOSs are not designed small. Figure 16 shows the VCDL delay-current characteristics by using variable delay cell. As the size of the delay cell is larger, the sensitivity at the necessary delay point is smoother.

Figures 17, 18, and 19 show the postlayout simulation results of the VCDL delay-current characteristics for DLL1, DLL2, and DLL3, respectively. The VCDL for DLL1 can achieve a target delay of 8.3 ns at about  $9 \mu\text{A}$  under variable conditions. The VCDL for DLL2 can achieve a target delay of 1.04 ns at  $80 \mu\text{A}$  under typical and the best conditions, which are tt/1.20 V/25°C and ff/1.35 V/−40°C, respectively. However, under the worst condition, which is ss/1.05 V/125°C,

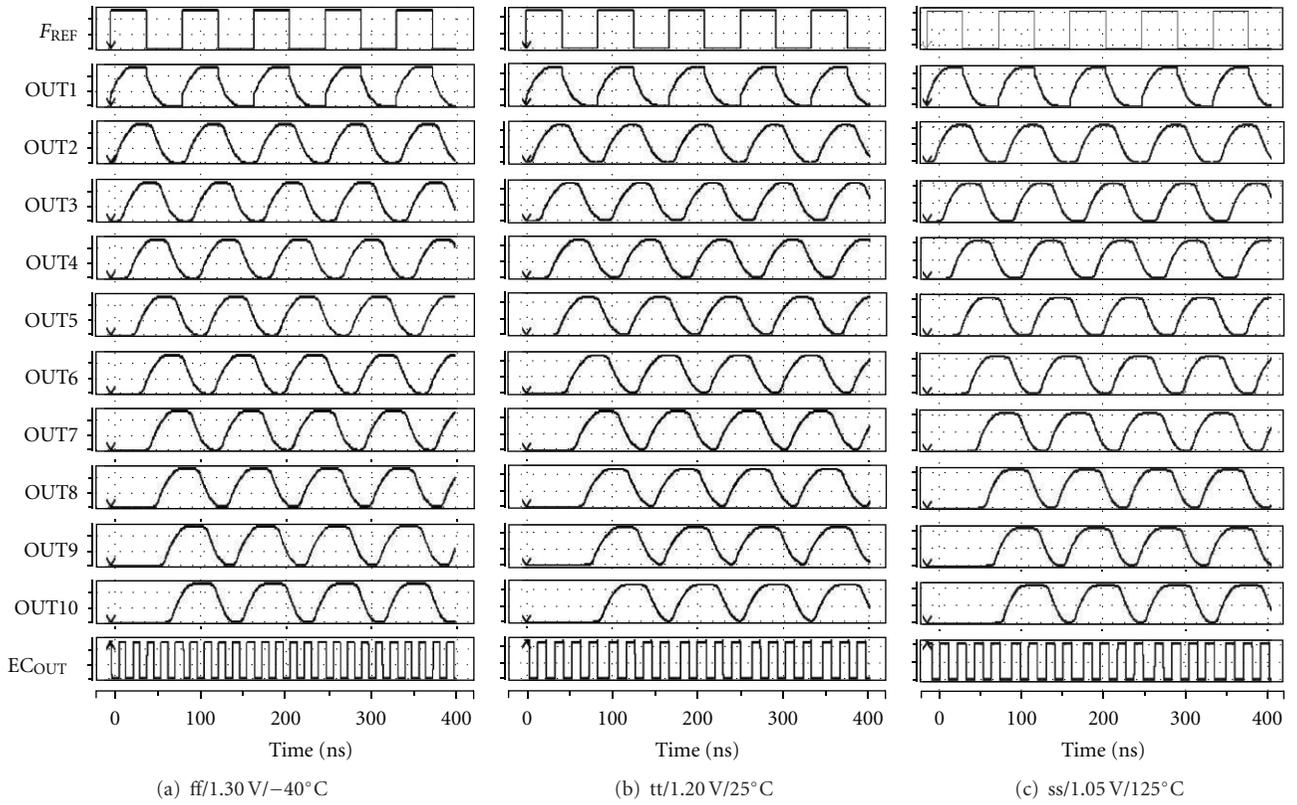


FIGURE 23: Postlayout simulation results from VCDL and EC applied to DLL1 for variable condition at  $V_C = 0.5$  V.

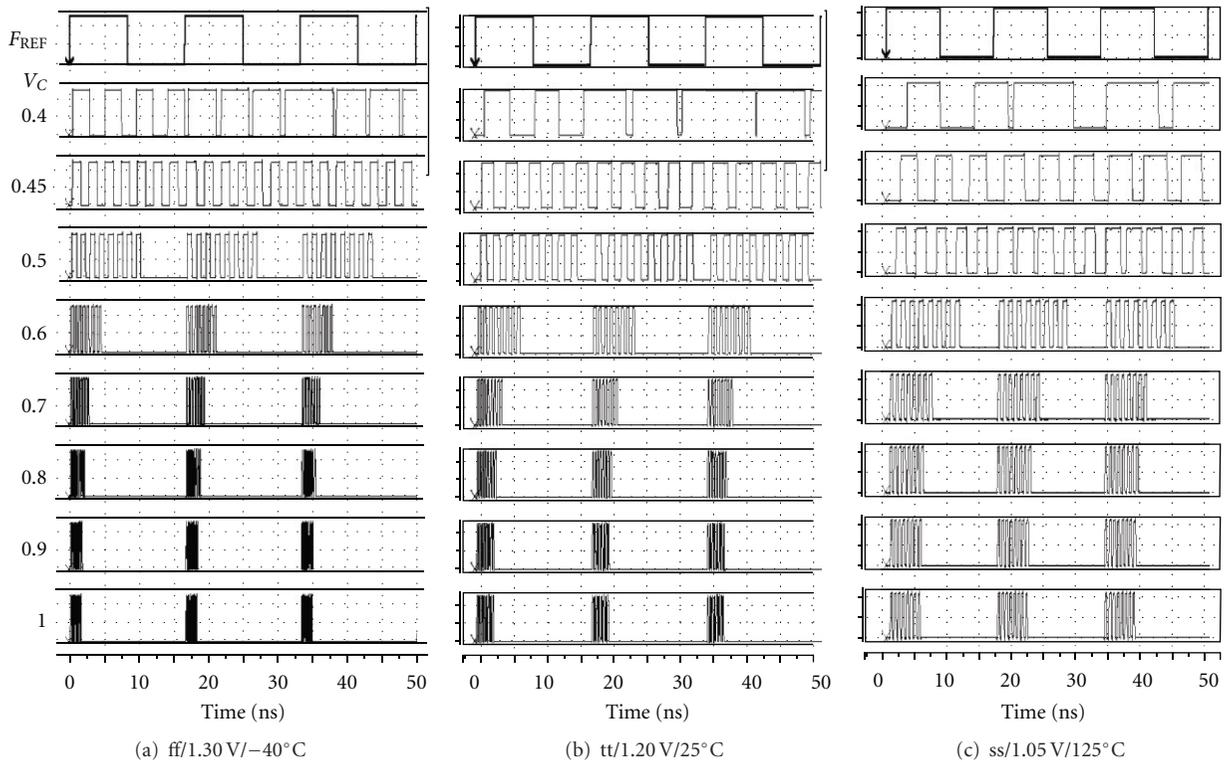


FIGURE 24: Postlayout simulation results from VCDL and EC applied to DLL2 for variable  $V_C$ .

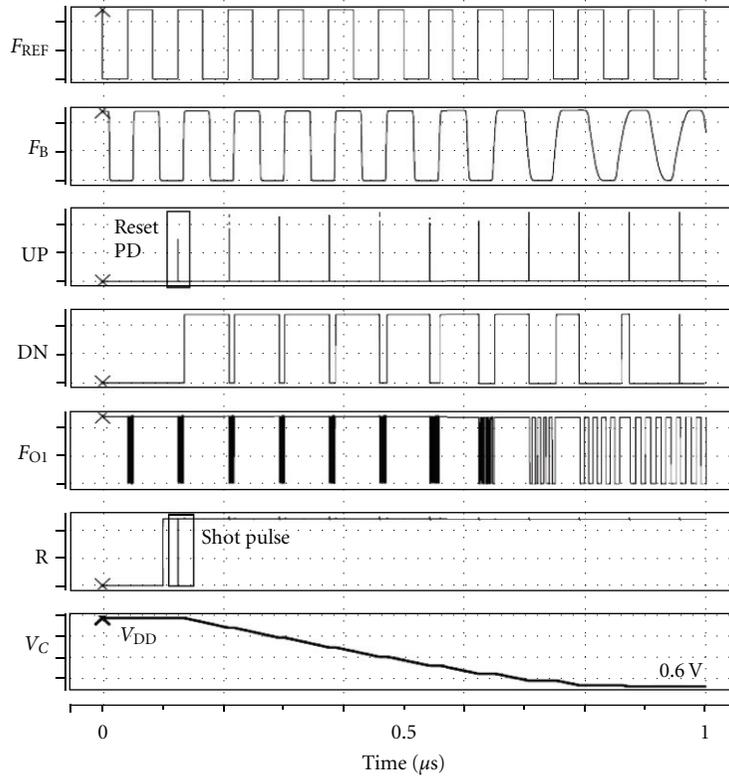


FIGURE 25: Postlayout simulation results from DLL1 locking operation. The simulation condition is tt/1.2 V/25°C.

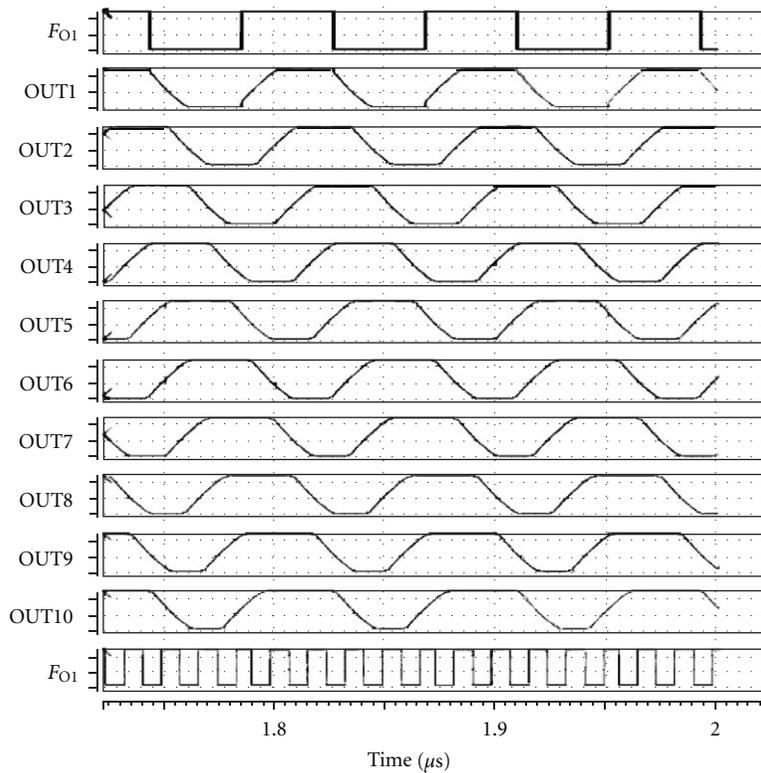


FIGURE 26: Postlayout simulation result of VCDL output signals from DLL1 settling operation. The simulation condition is tt/1.2 V/25°C.

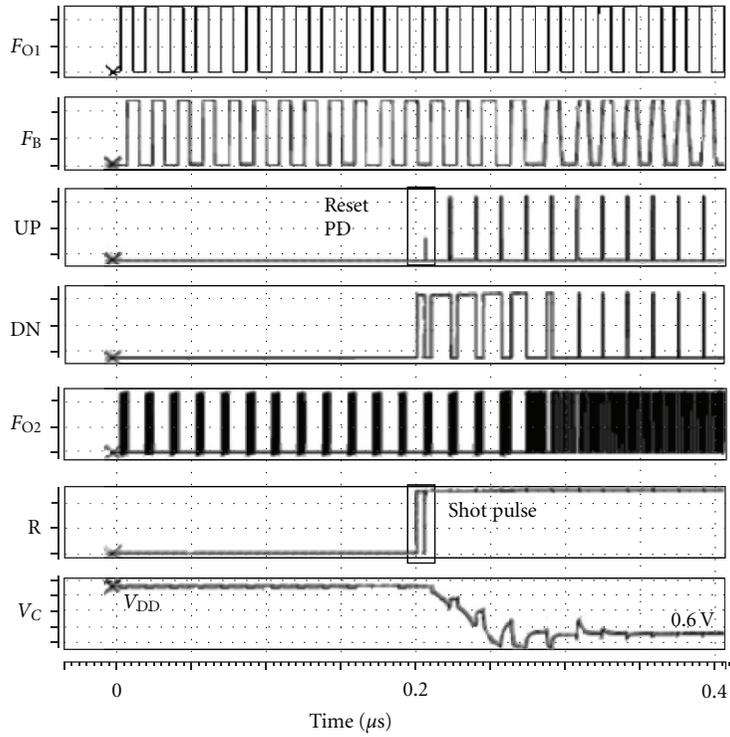


FIGURE 27: Postlayout simulation results from DLL2 locking operation. The simulation condition is  $tt/1.2 V/25^{\circ}C$ .

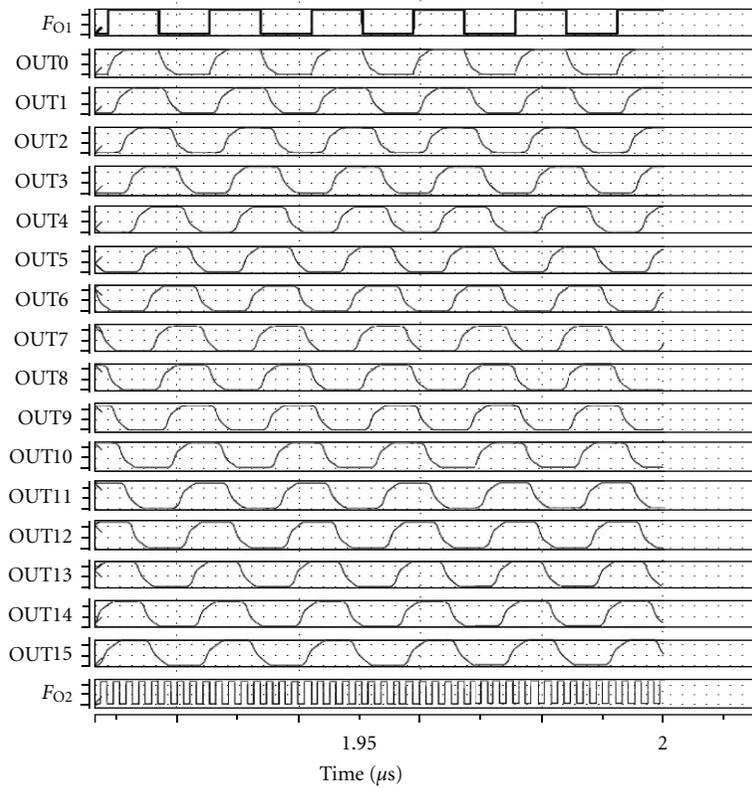


FIGURE 28: Postlayout simulation result of VCDL output signals from DLL2 settling operation. The simulation condition is  $tt/1.2 V/25^{\circ}C$ .

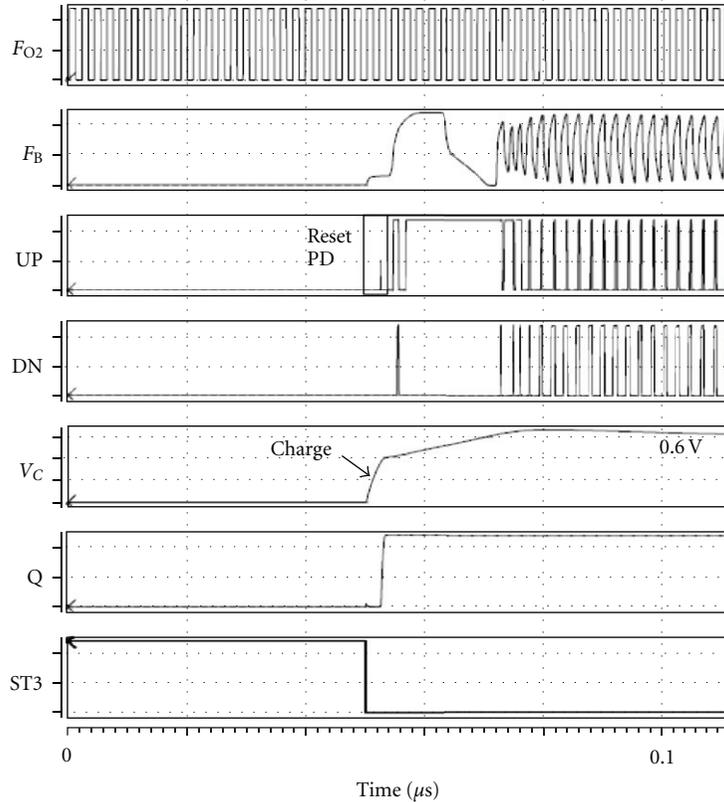


FIGURE 29: Postlayout simulation results from DLL3 locking operation. The simulation condition is  $tt/1.2\text{ V}/25^\circ\text{C}$ .

the target delay can be achieved at an  $I_C$  of about  $120\ \mu\text{A}$ . This is adjusted by the trimming bits (TRIM[1:0]), as shown in Figure 14. The VCDL for DLL3 can achieve a target delay of  $0.208\ \text{ns}$  at a  $V_C$  of about  $200\ \mu\text{A}$ . However, the VCDL is trimmed by the TRIM[1:0].

**3.4. Edge-Combiner.** Figure 20 shows a block diagram of the EC for DLL1. It consists of SR flip-flops (SRFFs) and NANDs and a NOR. Figure 21 shows a circuit diagram of the SRFF. It consists of six MOSs. There are two floating nodes (M3 drain node and M5 source node). Figures 22 and 23 show the postlayout simulation results from the VCDL and EC for DLL1. The one cycle delay is obtained at about  $V_C = 0.5\ \text{V}$  in  $ff/1.30\ \text{V}/-40^\circ\text{C}$  as shown in Figure 22(a). The EC can operate variable input signal as shown in Figure 22(b). If the SRFF cannot operate accurately by the leakage current, the output signal of the EC slips the clock in part. However, the EC can get all clock edges of the each signal at variable  $V_C$  as shown in Figure 22(b) and it can operate at variable conditions as shown in Figure 23.

Figure 24 shows the postlayout simulation results from the VCDL and EC for DLL2. The one cycle delay is obtained at between  $0.45\ \text{V}$  and  $0.50\ \text{V}$  in  $ff/1.30\ \text{V}/-40^\circ\text{C}$  and  $tt/1.20\ \text{V}/25^\circ\text{C}$  as shown in Figures 23(a) and 23(b), and at between  $0.50\ \text{V}$  and  $0.60\ \text{V}$  in  $ss/1.05\ \text{V}/125^\circ\text{C}$ . The EC can get all clock edges of the each signal at variable  $V_C$  and it can operate at variable conditions as shown in Figure 24.

**3.5. Lock Operation.** Figure 25 shows the postlayout simulation results of the DLL1 locking operation. The simulation condition is  $tt/1.2\ \text{V}/25^\circ\text{C}$ . The DLL1 has a capacitor of  $10\ \text{pF}$ . After ST1 is set to low at about  $100\ \text{ns}$ , the R is set to high, and then a shot pulse occurs. The PD operation is reset by the shot pulse, as shown by the UP and DN signals in Figure 25. After that, the PD generates a wide DN pulse and then the  $V_C$  decreases. Finally, DLL1 completes the lock at about  $1\ \mu\text{s}$ . When DLL1 completes the lock, the  $V_C$  is about  $0.6\ \text{V}$ . Figure 26 shows the VCDL output signals after the DLL1 completes the lock. The EC can generate the output signal ( $F_{O1}$ ) of  $60\ \text{MHz}$ .

Figure 27 shows the postlayout simulation result of the DLL2 locking operation. The simulation condition is  $tt/1.2\ \text{V}/25^\circ\text{C}$ . The DLL2 has a capacitor of  $0.5\ \text{pF}$ . After ST2 is set to low at about  $200\ \text{ns}$ , the R is set to high and then a shot pulse occurs. The PD operation is reset by the shot pulse, as shown by the UP and DN signals in Figure 27. After that, the PD generates a wide DN pulse and then the  $V_C$  decreases. Finally, DLL2 completes the lock at about  $400\ \text{ns}$ . When DLL2 completes the lock, the  $V_C$  is about  $0.6\ \text{V}$ . Figure 28 shows the VCDL output signals after the DLL2 completes the lock. The EC can generate the output signal ( $F_{O2}$ ) of  $480\ \text{MHz}$ .

Figure 29 shows the postlayout simulation results of the DLL3 locking operation. The simulation condition is  $tt/1.2\ \text{V}/25^\circ\text{C}$ . The DLL2 has a capacitor of  $1\ \text{pF}$ . After ST3 is set to low at about  $50\ \text{ns}$ , the PD operation is reset by

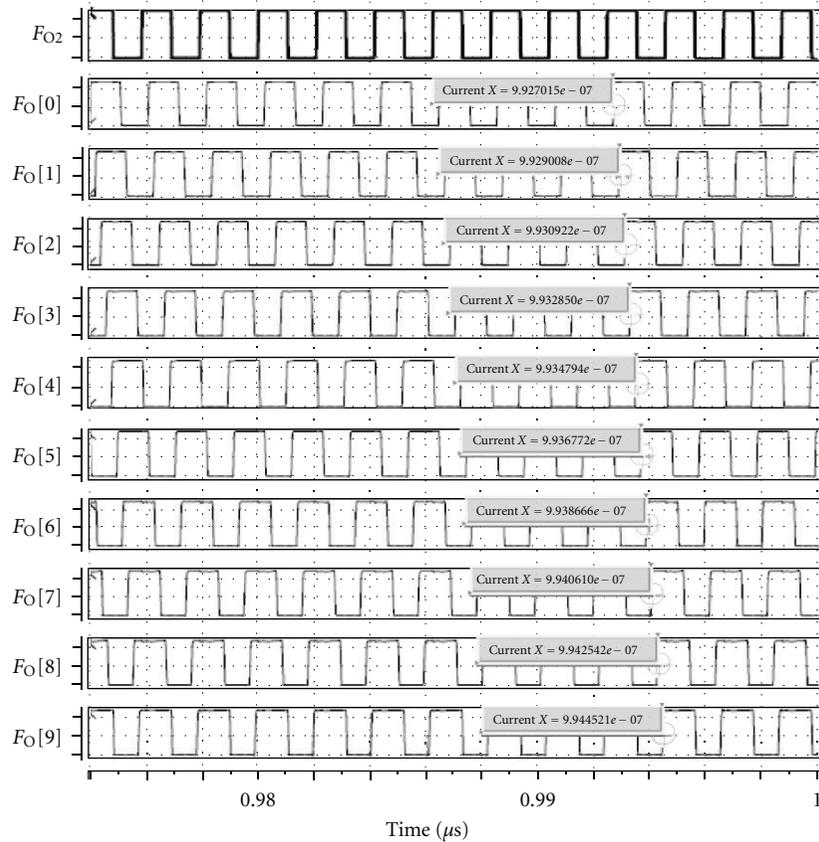


FIGURE 30: Postlayout simulation result of VCDL output signals from DLL3 settling operation. The simulation condition is  $tt/1.2\text{ V}/25^\circ\text{C}$ .

the R shot pulse, as shown by the UP and DN signals in Figure 28. After that, the PD generates a wide UP pulse and then the  $V_C$  increases. Finally, DLL3 completes the lock at about 100 ns. When DLL3 completes the lock, the  $V_C$  is about 0.6 V. Figure 30 shows the VCDL output signals after the DLL3 completes the lock. The DLL3 can generate the output signals of 480 MHz.

Figure 31 shows the postlayout simulation results from a clock generator that consists of DLL1, DLL2, and DLL3. After ST is set to low, ST1 is set to low first. At this time, ST2 and ST3 remain high.  $F_{REF}$  inserts DLL1 and the  $V_C$  is nearly  $V_{DD}$  due to a precharge. The PD generates a wide DN pulse at first because of the precharge. The  $V_C$  decreases due to the wide DN pulse. At about  $2\ \mu\text{s}$ , DLL1 completes the lock and generates  $F_{O1}$ , which is the 60 MHz clock signal. ST2 is set to low at about  $1\ \mu\text{s}$ . It is essentially set to low after DLL1 completes the lock. However, in this simulation, it is set to low before the DLL lock time. The  $V_C$  in DLL2 is almost  $V_{DD}$  due to the precharge. After ST2 is set to low, the PD in DLL2 generates a wide DN pulse and then the  $V_C$  is soon almost 0.5 V. At about  $2\ \mu\text{s}$ , DLL2 completes the lock. ST3 is set to low at about  $2\ \mu\text{s}$ . It is essentially set to low after DLL2 completes the lock, but in this simulation, it is set to low before the DLL lock time, too. After ST3 is set to low, the  $V_C$  at first remains almost  $V_{DD}$ . The  $V_C$  decreases at about  $2.6\ \mu\text{s}$  and finally is about 0.5 V. DLL3 completes the lock and generates the 10-tap 480 MHz clock signals. The total lock

time of the clock generator is about  $3.0\ \mu\text{s}$ . In general, the locking time of the DLL is defined by the capacitance and CP current. When the CP current is large for the capacitance, the locking time is short, but the locking operation is barely stable. DLL1 is designed to be stable because the phase error of DLL1 directly influences the other DLLs. DLL2 and DLL3 are designed to achieve a fast locking time because the clock generator can achieve it. Then, DLL2 and DLL3 start to operate before the forward DLL completes the fast locking time. When DLL2 starts to operate,  $F_{O1}$  is almost 60 MHz. Thus, DLL2 can accurately operate.

#### 4. Measurement Results

A 90 nm CMOS process was used to fabricate our proposed clock generator for use as a USB 2.0 PHY. Figure 32 shows the measurement results of the output signal  $F_{O[9]}$ . The measurement signal is  $F_{O[9]}$  divided by eight. The clock generator output signal frequency is 480 MHz. The jitter is less than 0.8 psrms. Figure 33 shows the measurement results of the EYE pattern for the USB 2.0 specifications. The USB 2.0 PHY with our proposed clock generator can pass these specifications. Figure 34 shows the measurement results of the random data pattern in the USB 2.0 specifications. The USB 2.0 PHY with our proposed clock generator can operate random data that meets the USB 2.0 specifications. Figure 35 shows the layout of the chip. Our proposed clock

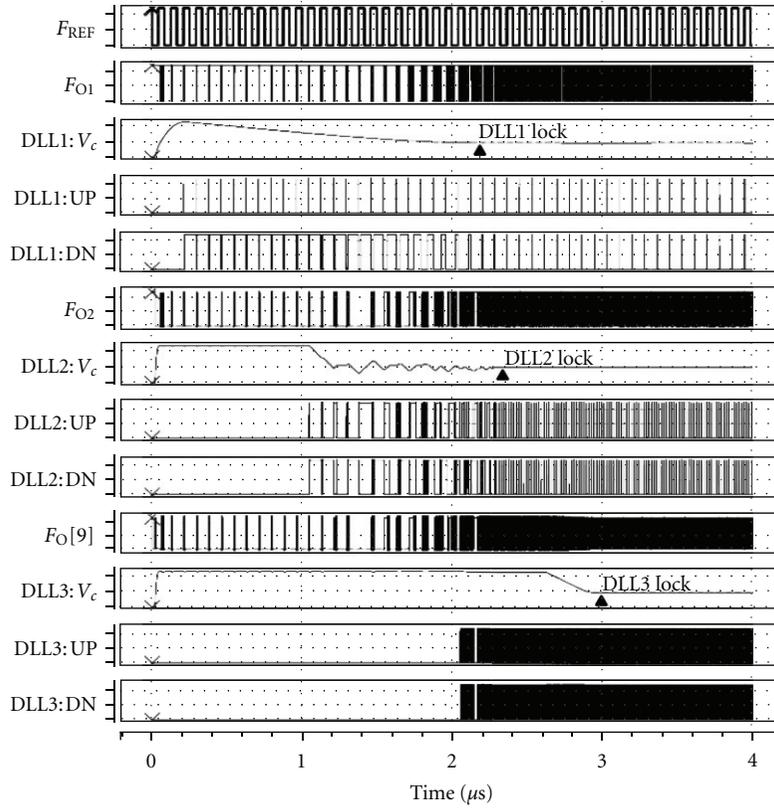


FIGURE 31: Postlayout simulation results from clock generator wake-up operation. The simulation condition is  $t_t/1.2\text{ V}/25^\circ\text{C}$ .

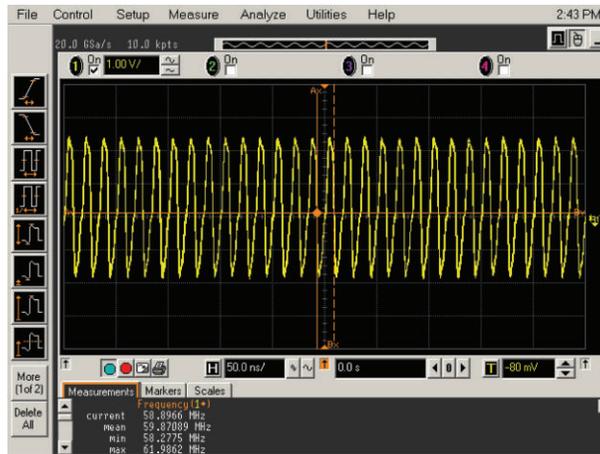


FIGURE 32: Measurement results from output signal  $F_{O[9]}$ . The measured signal is  $F_{O[9]}/8$ .

TABLE 1: Comparison table.

Item	Unit	Proposed clock generator	Conventional PLL in Figure 2
Structure	—	2 ECDLL + 1 DLL	1 PLL
Frequency	MHz	480	480
Locking-time	μs	3.5	10.0
Output jitter	ps	0.8	2.0
Technology	nm	90	90
Power	mW	1.3	3.8
Area	μm <sup>2</sup>	200 × 225	200 × 500

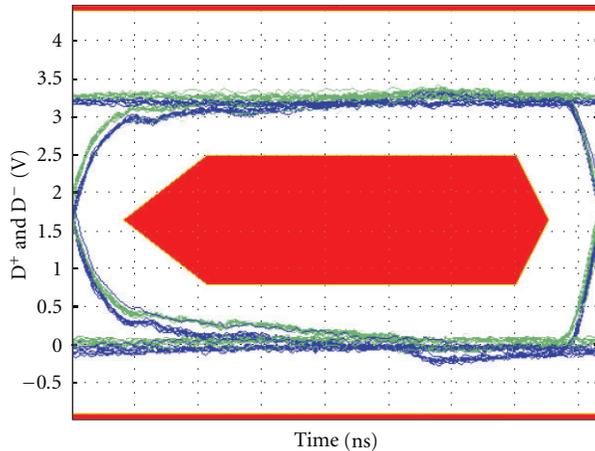


FIGURE 33: Measurement results from EYE pattern of fabricated PHY. The results meet the USB 2.0 specifications.

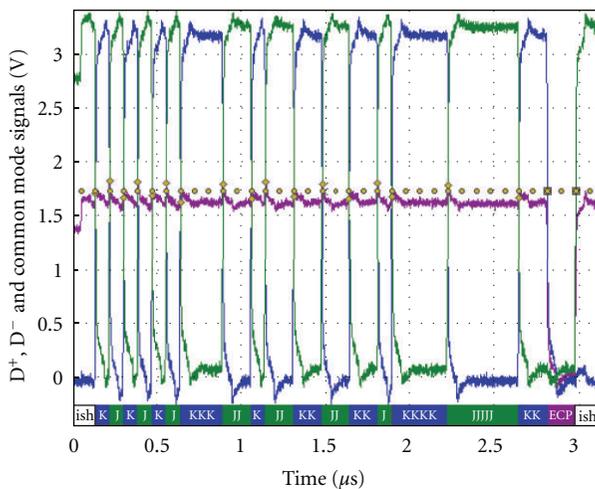


FIGURE 34: Measurement results from random data pattern of USB 2.0 specifications.

generator consists of three DLLs that is half the design area as that of the conventional one that consists of the PLL. Our clock generator consists of three DLLs. However, each DLL has a small capacitor to maintain the loop stability. Thus, our clock generator is smaller than the conventional one that has a large capacitor in the loop filter. Table 1 is a comparison table. The proposed clock generator has a power consumption of 1.3 mW, which is less than that of the conventional one, which is based on the PLL as shown in Figure 2. The ECDLL operates at the necessary reference signal frequency in the DLL loop that includes the VCDL. Thus, the power consumption is less than that of a PLL that has a VCO and a divider. A locking time of less than  $3.5 \mu\text{s}$  can also be achieved.

## 5. Conclusion

We proposed novel clock generator architecture to shrink the design area. The proposed clock generator consists of

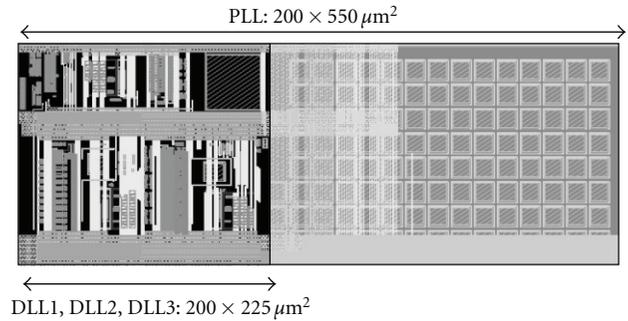


FIGURE 35: Chip layout. Our fabricated DLL occupies  $200 \times 225 \mu\text{m}$ .

two edge-combiner DLLs and a DLL. A shot pulse generator is used in the DLLs to prevent from harmonic lock and a CP with common-mode feedback is used in the DLLs to reduce the pattern jitter due to a constant phase error. A controller is used to control the wake-up sequence to prevent malfunctions. Our proposed clock generator is fabricated using a 90 nm CMOS process. It can achieve 10-tap 480 MHz clock signals that meet the USB 2.0 specifications. A power consumption of less than 1.3 mA was also achieved. Our USB 2.0 PHY with this clock generator also meets the USB 2.0 specifications. Our proposed clock generator needs only half the design area of the conventional one, which is based on the PLL.

## References

- [1] Y. Moon, G. Ahn, H. Choi, N. Kim, and D. Shim, "A quad 6 Gb/s multi-rate CMOS transceiver with TX rise/fall-time control," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '06)*, pp. 79–84, February 2006.
- [2] T. Kawamoto, T. Takahashi, S. Suzuki, T. Noto, and K. Asahina, "Low-jitter fractional spread-spectrum clock generator using fast-settling dual charge-pump technique for serial-ATA application," in *Proceedings of the 35th European Solid-State Circuits Conference (ESSCIRC '09)*, pp. 380–383, September 2009.
- [3] W. Grollitsch, R. Nonis, and N. Da Dalt, "A 1.4psrms-period-jitter TDC-less fractional-N digital PLL with digitally controlled ring oscillator in 65 nm CMOS," in *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC '10)*, pp. 478–479, February 2010.
- [4] "USB 2.0 transceiver macrocell interface (UTMI) specification version 1.05 3/29/2001," 2001, [http://developer.intel.com/technology/usb/download/2\\_0\\_Xcvr\\_Macrocell\\_1.05.pdf](http://developer.intel.com/technology/usb/download/2_0_Xcvr_Macrocell_1.05.pdf).
- [5] G. Chien and P. R. Gray, "A 900 MHz local oscillator using a DLL-based frequency multiplier technique for PCS applications," in *Proceedings of the IEEE International Solid-State Circuits Conference 47th Annual ISSCC*, pp. 202–203, February 2000.
- [6] K. H. Cheng, S. M. Chang, Y. L. Lo, and S. Y. Jiang, "A 2.2 GHz programmable DLL-based frequency multiplier for SOC applications," in *Proceedings of the IEEE Asia-Pacific Conference on Advanced System Integrated Circuits*, pp. 72–75, August 2004.
- [7] H. Y. Huang and J. H. Shen, "A DLL-based programmable clock generator using threshold-trigger delay element and circular edge combiner," in *Proceedings of the IEEE Asia-Pacific*

*Conference on Advanced System Integrated Circuits*, pp. 76–79, August 2004.

- [8] K. H. Cheng, S. M. Chang, S. Y. Jiang, and W. B. Yang, “A 2 GHz fully differential DLL-based frequency multiplier for high speed serial link circuit,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 2, pp. 1174–1177, May 2005.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

