High-Resolution All-Digital Duty-Cycle Corrector in 65-nm CMOS Technology

Ching-Che Chung, *Member, IEEE*, Duo Sheng, *Member, IEEE*, and Sung-En Shen

Abstract—In high-speed data transmission applications, such as double data rate memory and double sampling analog-to-digital converter, the positive and negative edges of the system clock are utilized for data sampling. Thus, these systems require an exact 50% duty cycle of the system clock. In this paper, two wide-range all-digital duty-cycle correctors (ADDCCs) with output clock phase alignment are presented. The proposed phase-alignment ADDCC (PA-ADDCC) not only achieves the desired output/input phase alignment, but also maintains the output duty cycle at 50% with a short locking time. In addition, the proposed high-resolution ADDCC (HR-ADDCC) without a half-cycle delay line can improve the delay resolution and mitigate the delay mismatch problem in a nanometer CMOS process. Experimental results show that the frequency range of the proposed ADDCCs is 263–1020 MHz for the PA-ADDCC and 200–1066 MHz for the HR-ADDCC with a DCC resolution of 3.5 and 1.75 ps, respectively. In addition, the proposed PA-ADDCC and HR-ADDCC are implemented in an all-digital manner to reduce circuit complexity and leakage power in advanced process technologies and, thus, are very suitable for system-on-chip applications.

Index Terms—All-digital duty-cycle corrector (ADDCC), delay-locked loop (DLL), digitally controlled delay line (DCDL), high resolution, phase alignment.

I. INTRODUCTION

DUTY-cycle correctors (DCCs) are widely used in high-speed devices, such as double data rate (DDR) memories, double sampling analog-to-digital converters, and system-on-a-chip (SoC) applications. Because both the positive and negative edges of the clock are utilized for sampling the input data, these systems require an exact 50% duty cycle of the input clock to ensure that the system meets the timing constraints. However, as the clock signal is distributed over the entire chip with clock buffers, the duty cycle of the clock is often far from 50% because of the unbalanced rise and fall times of the clock buffers, as a result of variations in process, voltage, and temperature. To resolve this problem, many approaches to correct the duty-cycle error and meet system requirements are proposed; for example, an analog pulse-width control loop
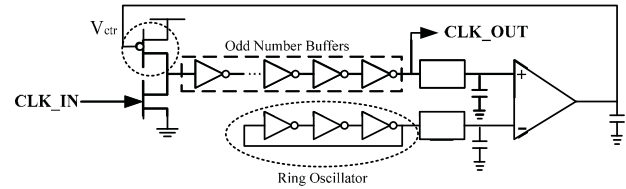
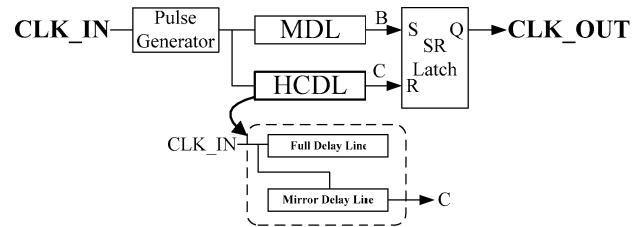Fig. 1. Conventional analog PWCL architecture [1].

Fig. 2. Block diagram of SMD-based DCC [10].

(PWCL) [1]–[4], [24], an all-digital PWCL [5]–[7], and an all-digital duty-cycle corrector (ADDCC) [8]–[12]. In addition, in some applications, the DCC is combined with delay-locked loop (DLL) located on the output side [13]–[18] to eliminate the phase error caused by the DCC circuit.

A conventional analog PWCL [1] uses a feedback approach to adjust the duty cycle of the input clock, as shown in Fig. 1. Based on the architectural requirements, it requires a ring oscillator to provide a 50% duty-cycle reference clock, and thus, the operating range and the acceptable input duty-cycle error are very restricted in this architecture [1]–[3]. A high linearity PWCL [4] that employs a linear control stage and a digitally controlled charge pump is proposed for extending the range of both input and output duty cycles over a wide frequency range. However, an analog PWCL usually requires a large on-chip capacitor that occupies a large chip area. In addition, the analog PWCL has a relatively longer lock-in time, and the leakage current problem of the charge pump makes it unsuitable for use in a nanometer CMOS process.

In contrast to PWCLs, all-digital PWCL and ADDCC do not utilize any passive components and use digital design approaches, making their integration into digital and low-supply voltage systems easy [5]–[12]. There are two major types of architecture in the digital approach: synchronous-mirror-delay (SMD) and time-to-digital converter (TDC). Fig. 2 shows the block diagram of the SMD-based DCC that consists of a half-cycle delay line (HCDL) and a match delay line [10]. The HCDL consists of a full-cycle delay line and a
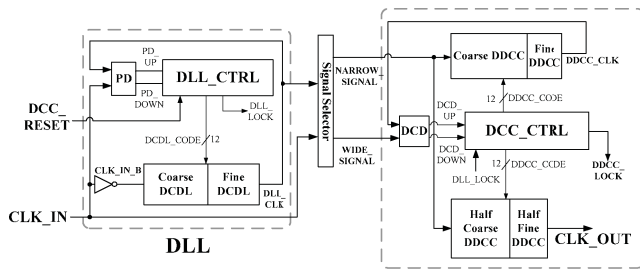
Fig. 3. Proposed PA-ADDCC.

Fig. 4. Timing diagram of DLL in the PA-ADDCC.

mirror delay line. The full-cycle delay line is used for detecting CLK_IN's period information, and the mirror delay line then generates the half-cycle delay time according to CLK_IN's period information. Subsequently, the 50% duty cycle clock is generated by an SR latch [5], [6], [8]–[11]. However, the two-delay-line architecture has a delay mismatch problem, particularly in the nanometer CMOS process with on-chip variations (OCVs). In addition, a high-resolution delay line is required for reducing the output duty-cycle error. Hence, the operating frequency range and the final duty-cycle error are limited in this architecture.

The TDC-based DCC quantizes CLK_IN's period information into a digital code, and then this digital code is divided by two to control the delay line for generating the half-cycle delay time [8], [9], [11], [12], [14], [17], [18]. The TDC-based DCC has a short locking time. However, the output digital code is divided by two from the quantization digital code, and thus, the duty-cycle correction resolution worsens along with the TDC quantization error. In addition, it is difficult to design a high-resolution TDC while maintaining a wide operation frequency range with low power and a small chip area. Although the TDC resolution has improved in recent years [19]–[21], this architecture is still not suitable for applications with a wide operating-frequency range.

Although the DCC can generate an output clock with a 50% duty cycle, the phases of the input and output clocks are misaligned. To reduce the phase error between the input and output clocks, the DCC is combined with a DLL located on the output side [13]–[18]. However, the integration of the DCC and DLL is a design challenge in these approaches. In addition, the power consumption and circuit complexity of the DLL have to be minimized to reduce the overall power and hardware costs.

In this paper, two wide-range ADDCCs with an output clock phase alignment are presented. First, the proposed phase-alignment ADDCC (PA-ADDCC) that consists of a DCC and a DLL not only achieves the desired output/input phase alignment but also maintains the output duty cycle at 50% with a short locking time. Second, the proposed high-resolution ADDCC (HR-ADDCC) uses a novel correction method without a HCDL to improve the delay resolution and mitigate the delay mismatch problem in a nanometer CMOS process. As compared with the SMD-based and TDC-based ADDCCs, the proposed designs can achieve high duty-cycle correction resolution and a wide operating frequency range easily while maintaining the phase alignment.
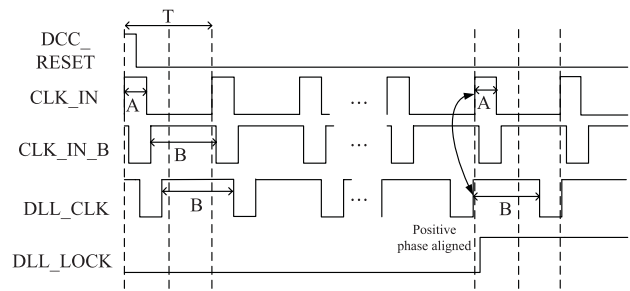
The rest of this paper is organized as follows. The architecture of PA-ADDCC and HR-ADDCC is presented in Section II. Section III describes the circuit implementation of the proposed designs. Section IV shows the measurement results of the PA-ADDCC test chip and the simulation results of the HR-ADDCC. Finally, the conclusion is given in Section V.

II. PROPOSED ADDCC ARCHITECTURE

The following sections present the two ADDCC architectures proposed in this paper: PA-ADDCC and HR-ADDCC. In addition, a brief summary of their comparison is given.

A. Phase-Alignment ADDCC

Fig. 3 shows the block diagram of the proposed PA-ADDCC. It is composed of a DLL, a signal selector, and a DCC. The DLL consists of a phase detector (PD), a coarse-tuning digitally controlled delay line (coarse DCDL), a fine-tuning digitally controlled delay line (fine DCDL), and a DLL controller (DLL_CTRL). The DCC consists of a duty-cycle detector (DCD), a coarse-tuning digitally controlled duty-cycle correction delay line (coarse DDCC), a fine-tuning digitally controlled duty-cycle correction delay line (fine DDCC), a half coarse-tuning digitally controlled duty-cycle correction delay line (half coarse DDCC), a half fine-tuning digitally controlled duty-cycle correction delay line (half fine DDCC), and a DCC controller (DCC_CTRL).

The inverted input clock (CLK_IN_B) passes through the delay line of the DLL, and then DLL's output signal (DLL_CLK) is compared with the input clock (CLK_IN) to align the positive edges of DLL_CLK and CLK_IN. The PD of the DLL detects the phase error between CLK_IN and DLL_CLK and then outputs PD_UP/PD_DOWN control signals to the DLL_CTRL. Subsequently, DLL_CTRL adjusts the delay line control code (DCDL_CODE) to compensate for the phase error. When the positive-edge phase error between CLK_IN and DLL_CLK is eliminated, the DLL is locked. Thus, two clocks (i.e., CLK_IN and DLL_CLK) with the complementary duty cycles are generated.

Fig. 4 shows the timing diagram of the DLL. The period of CLK_IN is T. If the duty cycle of CLK_IN is A/T and the duty cycle of DLL_CLK is B/T, then the period T is equal to $(A + B)$. In addition, the pulse width difference between the CLK_IN and the DLL_CLK is ΔE ($=B - A$). After the
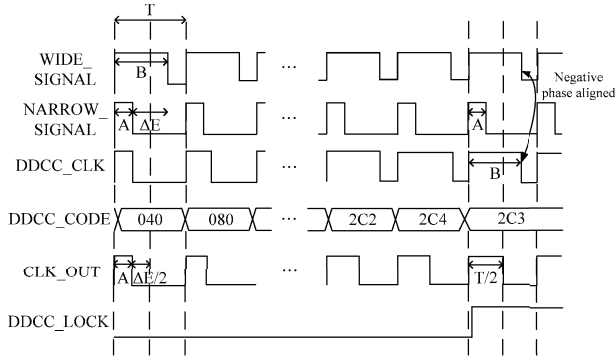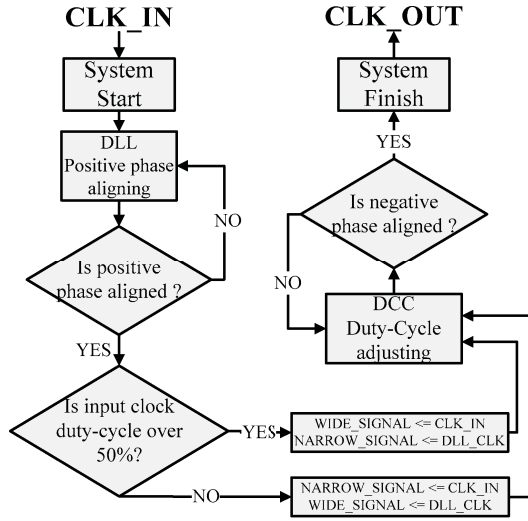
Fig. 5. Timing diagram of DCC in the PA-ADDCC.

Fig. 6. Operation flowchart of the PA-ADDCC.

Fig. 7. Proposed HR-ADDCC.

DLL is locked, the positive edges of CLK_IN and DLL_CLK are phase aligned. The signal selector detects the pulse widths of these two clocks, and the clock with a wider pulse width is output as WIDE_SIGNAL. In contrast, the clock with a shorter pulse width is output as NARROW_SIGNAL.

Fig. 5 shows the timing diagram of the DCC. After the DLL is locked, the proposed all-digital DCC starts to compensate the duty-cycle error ($\Delta E/2$) of the output clock (CLK_OUT). NARROW_SIGNAL passes through the coarse DDCC and the fine DDCC to increase the pulse width and is then output as DDCC_CLK. The DCD detects the negative-edge phase error between WIDE_SIGNAL and DDCC_CLK and then outputs DCD_UP/DCD_DOWN control signals to DCC_CTRL. Subsequently, DCC_CTRL adjusts the duty-cycle correction DDCC_CODE for increasing the pulse width of DDCC_CLK. When the negative-edge phase error is eliminated between WIDE_SIGNAL and DDCC_CLK, the DCC is locked.

The pulse width of NARROW_SIGNAL is increased by ΔE and outputs as DDCC_CLK. Since the period of CLK_IN is T, $(A+\Delta E/2)$ is equal to $T/2$ $(=A+(B-A)/2 = (A+B)/2)$. Therefore, the proposed DCC utilizes the half coarse DDCC and the half fine DDCC to increase the pulse width of CLK_IN by $\Delta E/2$. Therefore, after the DCC is locked, the duty cycle of CLK_OUT should be 50%. However, if there has duty-cycle
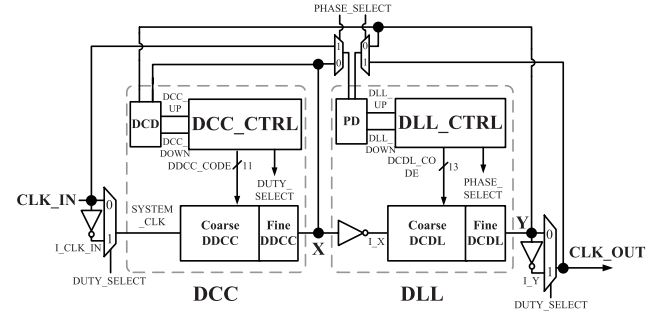
distortion caused by the coarse DCDL and the fine DCDL, the duty-cycle of CLK_OUT will have a residual duty-cycle error after the DCC is locked.

Fig. 6 shows the operation flowchart of the proposed PA-ADDCC. In the beginning, the DLL performs a positive-edge phase alignment. Until the positive edges are aligned, two clocks (i.e., CLK_IN and DLL_CLK) with complementary duty cycles are generated. At that time, if CLK_IN's duty cycle is more than 50%, CLK_IN and DLL_CLK are assigned to WIDE_SIGNAL and NARROW_SIGNAL, respectively. In contrast, if CLK_IN's duty cycle is < 50%, CLK_IN and DLL_CLK are assigned to NARROW_SIGNAL and WIDE_SIGNAL, respectively. These two signals are sent to DCC, and then NARROW_SIGNAL is compensated the duty cycle until its negative edge is aligned to WIDE_SIGNAL's negative edge. Then, CLK_OUT is generated, and the system is locked.

The proposed PA-ADDCC uses a sequential search with a high-resolution delay line, which can improve the accuracy of the duty-cycle correction as compared with the existing approaches. However, because of the half coarse DDCC and half fine DDCC inside it, PA-ADDCC may induce a delay mismatch problem when there are serious OCVs, particularly in an advanced CMOS process. Thus, HR-ADDCC is proposed for solving the possible delay mismatch problem in PA-ADDCC and for improving the accuracy of the duty-cycle correction further.

B. High-Resolution ADDCC

Fig. 7 shows the block diagram of the proposed HR-ADDCC. It is composed of an ADDCC and an all-digital DLL. The ADDCC consists of a DCD, a coarse-tuning digitally controlled duty-cycle correction delay line (coarse DDCC), a fine-tuning digitally controlled duty-cycle correction delay line (fine DDCC), and a DCC controller (DCC_CTRL). The all-digital DLL consists of a PD, a coarse DCDL, a fine DCDL, and a DLL_CTRL.

After the system is reset, both the DUTY_SELECT signal and the PHASE_SELECT signal are set to 0. The input clock (CLK_IN) is passed through DCC's delay line and output as an X signal. Subsequently, the inverted X signal is then passed through DLL's delay line and output as a Y signal. The PD of the DLL compares the phase error between the positive edges of X and Y, and then, it outputs DLL_UP/DLL_DOWN control
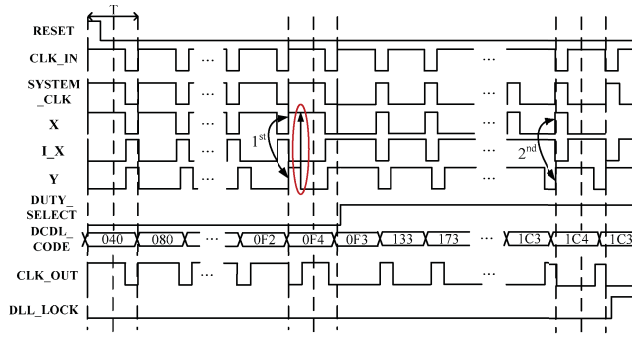
Fig. 8. Timing diagram of DLL when input clock's duty-cycle is over 50% in the HR-ADDCC.
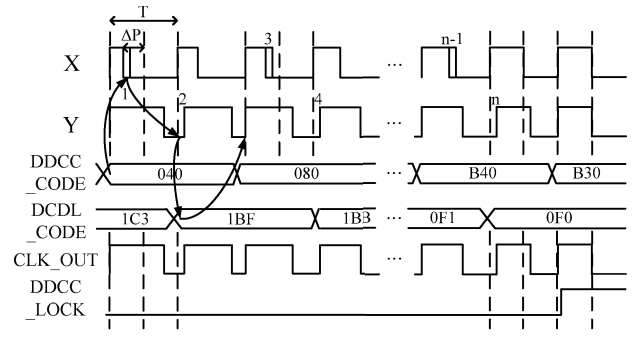

Fig. 9. Timing diagram of DCC when input clock's duty-cycle is over 50% in the HR-ADDCC.

Fig. 10. Timing diagram of DCC when input clock's duty-cycle is over 50% in the HR-ADDCC.

signals to DLL_CTRL. DLL_CTRL adjusts the DCDL_CODE to compensate for the phase error. When the phase error between X and Y is eliminated, the DLL is locked. Then, two clocks (i.e., X and Y) with complementary duty cycles are generated. Thus, if the period of the input clock (CLK_IN) is T and the duty cycles of X and Y are A/T and B/T, respectively, the period T is equal to $(A + B)$.

If CLK_IN's duty cycle is more than 50%, DLL needs a second relocking procedure for ensuring that the duty cycle of the X signal is always smaller than 50% before the DCC operation. Fig. 8 shows the DLL timing diagram when the input clock's duty cycle is more than 50%. After the first locking procedure, if the negative edge of the X signal lags behind the negative edge of the Y signal, which means the duty cycle of the input clock (CLK_IN) is larger than 50%, the DUTY_SELECT signal is set to 1, and therefore, the input clock is switched to inverted CLK_IN (I_CLK_IN) to guarantee that the duty cycle of the X signal is always < 50%. In addition, the output clock is switched to the inverted Y (I_Y) signal, and the DLL eliminates the phase error between CLK_IN and CLK_OUT.

After the DLL is locked, the proposed all-digital DCC starts to compensate for the duty-cycle error of the output clock (CLK_OUT). The DCD detects the phase error between the negative edges of X and Y, and then, it outputs DCC_UP/DCC_DOWN control signals to DCC_CTRL. DCC_CTRL adjusts the duty-cycle correction DDCC_CODE to increase the pulse width of the X signal according to the outputs of the DCD. Fig. 9 shows the timing diagram for the DCC operation when the input clock duty cycle is < 50%.

In the first cycle, the DCC extends the pulse width of the X signal. Then, in the next cycle, the positive edge of the Y signal lags behind the positive edge of the X signal because of the pulse extension in the previous cycle. Thus, in the second cycle, DCDL_CODE is decreased for aligning the positive edges of X and Y. The same process is repeated until both the positive edges and the negative edges of X and Y are phase aligned; then, the DCC is locked. For loop stability, in each clock cycle, only one loop (DCC or DLL) is working.

The pulse width of the X signal is increased by ΔP, which is equal to $(B - A)/2$. Since the period of CLK_IN is T, $(A + \Delta P)$ is equal to $T/2$ $(= A + (B - A)/2 = (A + B)/2)$. Therefore, after the DCC is locked, the duty cycle
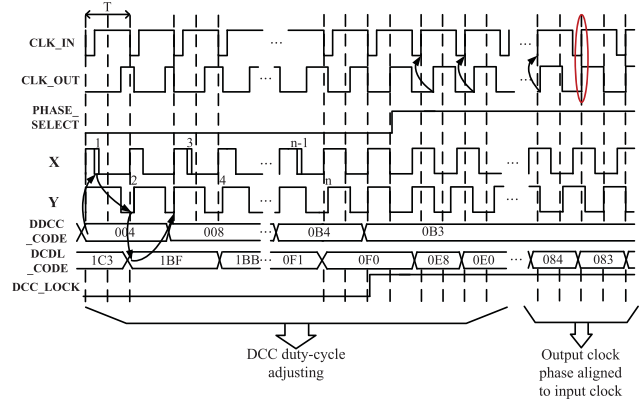
of CLK_OUT is 50%. Further, once the DCC is locked, the PHASE_SELECT signal is set to 1. The inputs of DLL's PD are switched to CLK_IN and CLK_OUT. Then, DLL adjusts DCDL_CODE to compensate for the phase error between CLK_IN and CLK_OUT. Therefore, the output clock (CLK_OUT) can be phase aligned with the input clock (CLK_IN).

Fig. 10 shows the DCC timing diagram when the input clock's duty cycle is more than 50%. After the DLL is locked, the DCC starts to correct the duty cycle of the output clock. Before the DCC is locked, the correction action is the same as in the case when the input clock's duty cycle is < 50%. Nevertheless, the positive edge of CLK_OUT is not aligned with the positive edge of CLK_IN. Therefore, we select I_Y as the CLK_OUT signal as the positive edge of inverted Y (I_Y) lags to CLK_OUT. Thus, DLL just reduces DCDL_CODE, which in turn reduces the delay time of DCDL until the positive edges between CLK_IN and CLK_OUT are aligned.

Fig. 11 shows the operation flowchart of the proposed HR-ADDCC. At the beginning, DLL performs a positive-edge phase alignment. When DLL is at the first time lock, DCC determines whether CLK_IN's duty cycle is < 50% or more. If CLK_IN's duty cycle is < 50%, then DCC duty cycle is adjusted. Otherwise, DCC sets DUTY_SELECT to 1, and the DLL phase alignment is carried out until the second time lock. When DLL is locked, we start the DCC duty-cycle adjustment.

Fig. 11.   Operation flowchart of the HR-ADDCC.

Fig. 12.   MUX-type coarse-tuning component of the proposed DCDL.

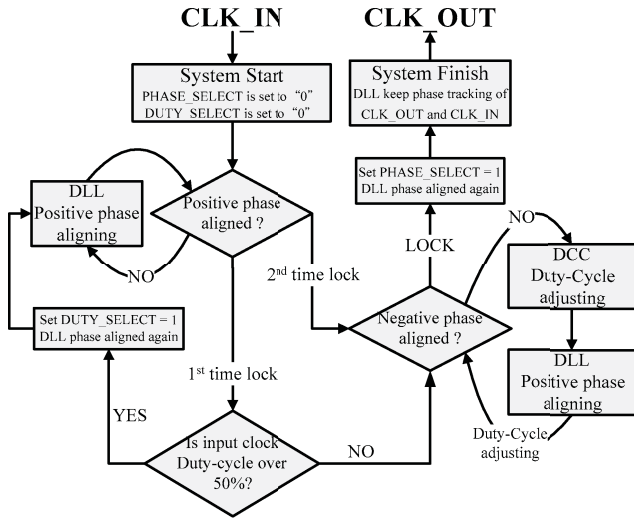In the first cycle, DCC extends the pulse width of the X signal. Then, in the next cycle, the positive edge of the Y signal lags behind the positive edge of the X signal because of the pulse extension in the previous cycle. Thus, in the second cycle, DLL aligns the positive edges of X and Y. The same process is repeated until both the positive edge and the negative edge of X and Y are phase aligned, and then, DCC is locked. After DCC is locked, DLL sets PHASE_SELECT to 1 and keeps tracking the phase between the output clock (CLK_OUT) and the input clock (CLK_IN).

C. Comparisons of the Two Proposed ADDCCs

The proposed PA-ADDCC employs a high-resolution phase tracking method that can improve the accuracy of 50% duty-cycle correction as compared with the existing approaches. However, the duty-cycle detection is carried out by the full-cycle delay line, and the output clock is compensated by the half-cycle delay line. The duty-cycle compensation code from the full-cycle delay line to the HCDL is shifted by one bit. Thus, the HCDL reduces the duty-cycle correction precision. In addition, if the detect circuit and the output circuit are not the same, there would be a delay mismatch problem with serious OCVs. In the proposed PA-ADDCC, the final duty-cycle error mainly comes from PD's dead zone, DCDL's resolution, DCC's DCD dead zone, DDCC's resolution, and the delay resolution of half DDCC. In addition, because the proposed PA-ADDCC has an intrinsic delay that comes from the signal selector and the half DDCC, there still has residual phase error after PA-ADDCC is locked. In contrast, the phase error between input and output clock can be reduced by a DLL after duty-cycle correction in the proposed HR-ADDCC.

In contrast to PA-ADDCC, the proposed HR-ADDCC also uses the high-resolution phase tracking method but eliminates the delay mismatch problem and half DDCC's resolution problem. Therefore, the output clock duty-cycle error mainly comes from DCDL's resolution, PD's dead zone, DDCC's resolution, and DCD's dead zone. Thus, HR-ADDCC can improve the accuracy of duty-cycle correction. According to
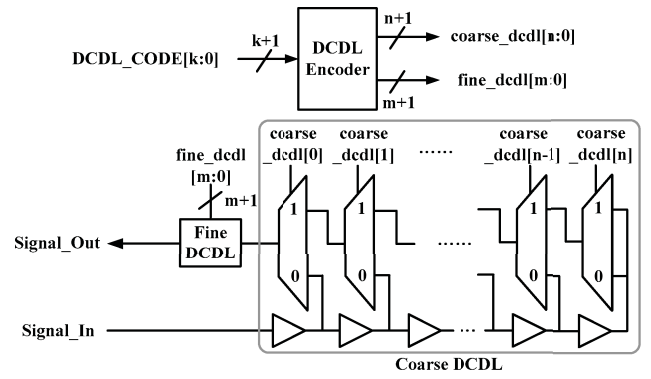
the experimental results, the duty cycle correction resolution of PA-ADDCC and HR-ADDCC is 3.5 and 1.75 ps, respectively.

Further, after the duty-cycle correction, the proposed HR-ADDCC can keep tracking the phase error between the external clock (CLK_IN) and the output clock (CLK_OUT). Thus, the output clock can deskew the phase error and reduce the intrinsic delay of the circuit. Nevertheless, the duty-cycle correction algorithm of HR-ADDCC has two steps in each duty-cycle correction operation, and thus, HR-ADDCC has a longer locking time than PA-ADDCC. For example, with a 1-GHz 20% duty-cycle input clock, the lock-in time for PA-ADDCC and HR-ADDCC is 24 and 120 cycles, respectively.

III. Circuit Implementation

The important components of the proposed ADDCC are presented in this section. The key functional blocks in DLL including a PD and a digitally controlled delay line (DCDL) are discussed first. Next, the DCD and the digitally controlled duty-cycle corrector (DDCC) delay line in the DCC are presented.

A. Key Components in DLL

Fig. 12 shows the multiplexer (MUX)-type DCDL architecture, which combines the coarse DCDL and the fine DCDL. The circuit operating path is from Signal_In to Signal_Out, which is selected by control code coarse_dcdl [n:0]. The coarse DCDL has $n + 1$ coarse-tuning delay cells, and each coarse-tuning delay cell is combined with a buffer and a multiplexer. Therefore, the coarse DCDL can provide $n + 1$ different delay times and easily cover a wide frequency range. Nevertheless, the coarse DCDL resolution is not sufficiently good, and thus, the fine DCDL is added to achieve a high-resolution DCDL.

Fig. 13 shows the fine-tuning component of DCDL, which is based on digitally controlled varactors (DCVs) [23]. Each DCV cell is composed of four NAND gates. The fine DCDL can provide $m + 1$ different delay times by controlling the DCV cells. When the control bit of a DCV cell is enabled, the capacitance at the output node of the inverter is changed and the delay time is increased accordingly. Thus, the resolution of DCDL can be improved by the use of fine DCDL.
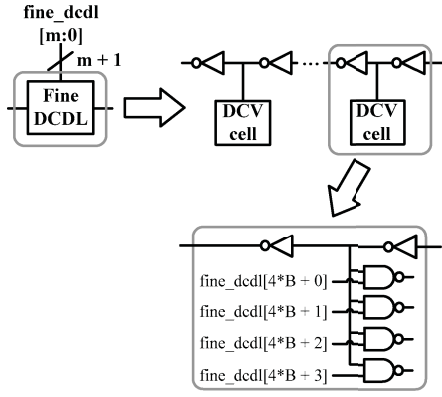
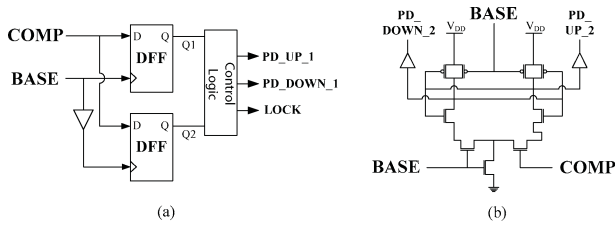Fig. 13. Fine-tuning component of the proposed DCDL.

Fig. 14. Proposed phase detector (PD). (a) Sampled-based PD. (b) Sense-amplifier-based PD [22].

According to the application requirements, the resolutions of coarse-tuning and fine-tuning components are 51 and 2.75 ps, respectively; the number of bits of coarse-tuning and fine-tuning components is 7 and 5, respectively.

Fig. 14 shows the proposed PD; it detects the positive-edge phase error between COMP and BASE. The proposed PD is composed of a sampled-based PD (SBPD) and a sense-amplifier-based PD [22]. To improve the detectable phase error, a sense-amplifier-based PD that can detect a small phase error larger than 1 ps in a typical case simulation is used in the PD design.

Although the sense-amplifier-based PD can detect a small phase error, it has incorrect detection results when the phase error between two inputs is large because of the leakage current of the transistor in the 65-nm CMOS process. For this reason, we use the SBPD to detect the large phase error at the beginning and then use the sense-amplifier-based PD to improve the overall phase error detection capability. Although the SBPD does not have a small dead zone because of the setup/hold time requirements of the D-Flip/Flops, the SBPD can be designed easily and can be built with standard cells. It can prevent the incorrect operation of the sense-amplifier-based PD.

At the beginning, the PD controller receives SBPD's outputs (PD_UP_1 and PD_DOWN_1). After the SBPD is locked, the PD controller switches to receive sense-amplifier-based PD's outputs (PD_UP_2 and PD_DOWN_2). Therefore, the proposed PD can correctly detect a small phase error between COMP and BASE.
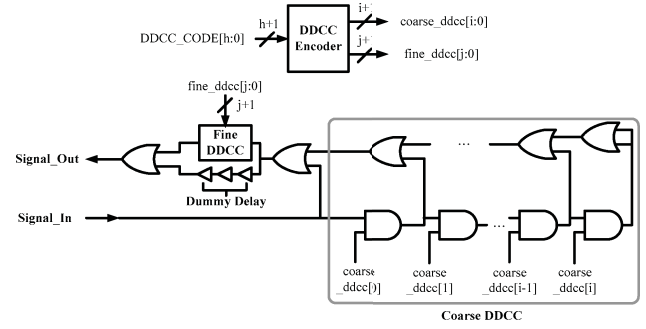

Fig. 15. AND-OR-Type coarse-tuning component of the proposed DDCC.

B. Key Components in DCC

Fig. 15 shows the AND–OR-type DDCC architecture, which is combined with the coarse-tuning component (coarse DDCC) and the fine-tuning component (fine DDCC). The circuit operating path is from Signal_In to Signal_Out, which is selected by control code coarse_ddcc [i:0]. The Coarse DDCC has $i + 1$ coarse-tuning delay cells, and each coarse-tuning delay cell is combined with an AND cell and an OR cell. Therefore, coarse DDCC can provide $i + 1$ types of pulse-width adjustments and easily cover the wide pulse-width adjustment range. Nevertheless, the resolution of coarse DDCC is not sufficiently good. For this reason, the fine-tuning component is added to increase the DDCC resolution.

The fine DDCC fine-tuning component is based on the architecture of DCV [23], which is the same as that of fine DCDL. Each DCV cell is combined with four NAND gates, which are controlled by enable code fine_ddcc[j:0]. The fine-tuning component has $j + 1$ types of delay times brought about by controlling the DCV cells. An OR gate is connected after the DCV output and the dummy delay output. The dummy delay is used for reducing DCV's intrinsic delay.

The architecture of the proposed DCD is similar to that of the proposed PD, as shown in Fig. 14. It also consists of a sample-based DCD and a sense-amplifier-based DCD. However, the proposed DCD detects the negative-edge phase error between COMP and BASE. Thus, in the proposed DCD, there are two inverters in front of PD's inputs (COMP and BASE). Then, PD can easily transform into the proposed DCD, and the operation behavior is same as that of the proposed PD.

IV. EXPERIMENTAL RESULTS

The proposed PA-ADDCC is fabricated using a standard performance (SP) 65-nm CMOS process, and the microphotograph of the PA-ADDCC is shown in Fig. 16. The active area is 100×100 μm^2, and the chip area with I/O pads is 644×644 μm^2. The test chip consists of a test circuit, a DLL, and a DCC. The input frequency ranges from 262 MHz to 1.02 GHz. Because the maximum input and output signal frequencies are restricted by the I/O pad speed limitation, which is approximately < 300 MHz, the test circuit needs to provide a high-frequency clock signal with a programmable duty cycle as the input clock to the ADDCC circuit. In addition, if the output clock frequency is higher than 300 MHz,

Fig. 16.   Microphotograph of the proposed PA-ADDCC.

Fig. 17.   Block diagram of test chip.

it should be divided into low frequencies before being output to the I/O pad.

Fig. 17 shows the block diagram of a PA-ADDCC test chip. The test chip is combined with a test mode control circuit (TEST_MODE_CTRL), a digitally controlled oscillator (DCO), a duty-cycle generator delay line (DUTY_DELAY_GEN), and a divider circuit (DIV_FOUR). The input FREQ_SELECT determines the DCO output frequency. DCO is designed on the basis of the MUX-type DCO architecture, and its output frequency ranges from 242 to 1094 MHz. The input DUTY_SELECT is encoded into DUTY_CODE, which determines the delay time from A to B, as shown in Fig. 17. Then, A and B are combined by an OR gate to adjust the duty cycle of the DCO output clock. If a duty cycle of $< 50\%$ is required, the inverted signal (I_C) is selected as the input clock to ADDCC. The ADDCC input clock (SYSTEM_CLK) can be an external low-frequency clock (INPUT_CLK) or the internal high-frequency clock (DCO_CLK).

The maximum output signal frequency is restricted by the I/O pad speed limitation, and thus, ADDCC's output clock (OUTPUT_CLK) is divided by four before output. Fig. 18 shows the block diagram of the DIV_FOUR circuit. CLK_P is triggered by CLK_IN's positive edge, and CLK_N is triggered by CLK_IN's negative edge.

Fig. 18.   Block diagram of DIV_FOUR circuit.

Fig. 19.   Timing diagram of DIV_FOUR circuit.

Fig. 19 shows the timing diagram of the DIV_FOUR circuit. If CLK_IN's period is T, then CLK_P's and CLK_N's period is TD ($= 4 \times T$). In addition, CLK_IN's duty cycle is A/T, and the phase difference between CLK_P's positive edge and CLK_N's positive edge is A. Therefore, we can measure the period of CLK_P and CLK_N to obtain TD, and the phase difference between CLK_P and CLK_N is A. Then, the duty cycle of CLK_IN can be calculated as A/(TD/4), which is equal to A/T. In addition, the enable-bit DIV_OPEN controls the output clock gating for saving power. If DIV_OPEN is set to 0, CLK_P and CLK_N are gated and ORIG_CLK is the enabled output. If DIV_OPEN is set to 1, ORIG_CLK is gated and CLK_P and CLK_N are the output.

Fig. 20 shows the duty-cycle measurement results of the proposed PA-ADDCC. In Fig. 20, signal #1 is CLK_P, and signal #4 is CLK_N. In Fig. 20(a), the phase difference between CLK_P and CLK_N is 1.898 ns, and the period of CLK_N is 15.241 ns; thus, the duty cycle of the ADDCC's output clock is 49.8% ($= 1.898/(15.241/4) \times 100\%$) at 262 MHz. Fig. 20(b) shows that the phase difference between CLK_P and CLK_N is 484.7 ps and the period of CLK_N is 3.92 ns; thus, the duty cycle of ADDCC's output clock is 49.4% ($= (0.4847/(3.92/4)) \times 100\%$) at 1020 MHz.

Fig. 21 shows the output clock duty-cycle measurement results of the proposed PA-ADDCC. The input clock frequency ranges from 262 to 1020 MHz, and the duty cycle of the input clock ranges from 14% to 86%. In addition, the core power supply voltage is 1.0 V, and the pad power supply voltage is 2.0 V. The power consumption is 6.5 mW (at 1.02 GHz), 2.68 mW (at 442 MHz), and 1.96 mW (at 262 MHz).

Fig. 22 shows the output clock jitter histogram of the proposed PA-ADDCC. The root mean square (rms) jitter and

(a)

(b)

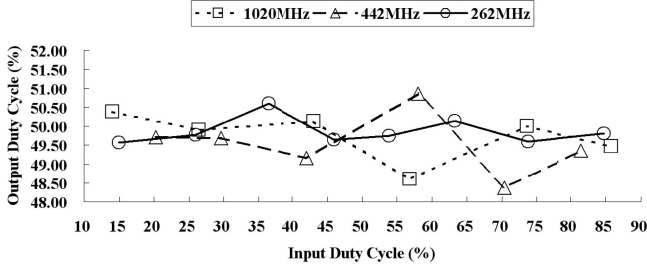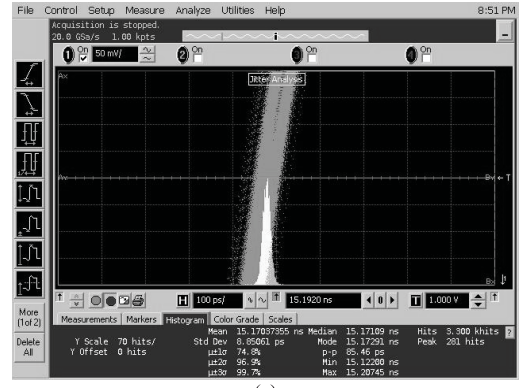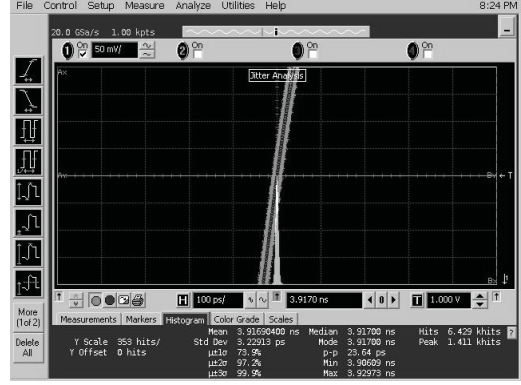Fig. 20. Duty-cycle measurement results of the proposed PA-ADDCC at (a) 262 MHz and (b) 1020 MHz.

(a)

(b)

Fig. 22. Jitter histogram of the proposed PA-ADDCC at (a) 262 MHz and (b) 1021 MHz.

Fig. 21. Measurement results of the proposed PA-ADDCC.
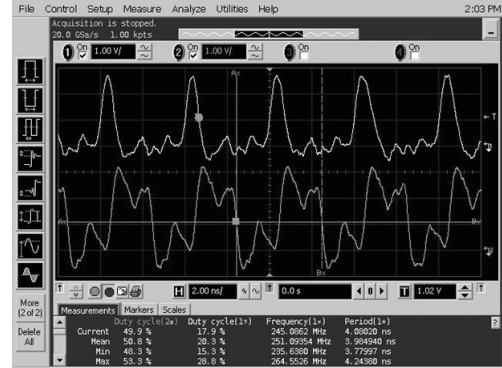
Fig. 23. 20% duty-cycle input clock and 50% duty-cycle output clock at 250 MHz.

peak-to-peak (pk–pk) jitter at 262 MHz are 8.85 and 85.46 ps, respectively. In addition, the rms jitter and the pk–pk jitter at 1021 MHz are 3.2 and 23.64 ps, respectively. Fig. 23 shows the measurement results for a 20% duty-cycle input clock and the 50% duty-cycle output clock at 250 MHz. In Fig. 23, signal #1 is the external input clock and signal #2 is the ADDCC output clock with a 50.8% duty cycle. Because the proposed PA-ADDCC has an intrinsic delay that comes from the delay of the signal selector and the intrinsic delay of the half DDCC, the output clock has residual phase error with the reference clock. In addition, the MUX of the test circuit and the output clock gating control circuit also causes an additional phase error in the measurement result shown in Fig. 23.

The proposed HR-ADDCC is implemented using a SP 65-nm CMOS process, and the layout of the HR-ADDCC is shown in Fig. 24. The active area is $100 \times 100 \ \mu m^2$, and

the chip area with I/O pads is $644 \times 644 \ \mu m^2$. The test chip consists of a test circuit, a DLL, and a DCC. Fig. 25 shows the simulation results of the proposed HR-ADDCC. The input clock frequency ranges from 200 MHz to 1.066 GHz, and the duty cycle of the input clock ranges from 20% to 80%.

Table I shows a performance comparison with prior works. In the case of TDC-based all-digital DCC architecture, it is difficult to achieve a high duty-cycle correction resolution while maintaining a wide operation frequency [6], [8], [12], [16]. Although analog PWCLs [4], [7] have precisely a 50% duty-cycle output, they require a large on-chip capacitor that occupy a large chip area, and they also have a long

TABLE I

PERFORMANCE COMPARISONS

Parameter	Proposed PA-ADDCC	Proposed HR-ADDCC	JSSC'06 [6]	TCAS-II'07 [12]	JSSC'09 [16]	VLSI'12 [8]	JSSC'08 [4]	JSSC'05 [7]	ESSCIRC'12 [24]
Type	Sequential Search/HCDL	Sequential Search	TDC	TDC	TDC/HCDL	TDC/HCDL	TDC/Analog PWCL	Analog PWCL	Analog dual-loop PWCL
Process	65 nm	65 nm	0.35 μm	0.18 μm	0.18 mm	0.18 μm	0.18 μm	0.35 μm	45 nm
Supply voltage	1.0 V	1.0 V	3.3 V	1.8 V	1.8 V	1.8 V	1.8 V	3.3 V	1.8 V + 1.1 V
Max. Frequency (MHz)	1020	1066	600	1200	1500	2000	1100	1270	2000
Min. Frequency (MHz)	262	200	400	800	440	400	50	1000	1000
Input Duty Cycle Range	14% ∼ 86%	20% ∼ 80%	30% ∼ 70%	40% ∼ 60%	40% ∼ 60% @1.5 GHz	10% ∼ 90% @400 MHz 20% ∼ 80% @2 GHz	30% ∼ 70%	N/A	25% ∼ 75%
Output 50% Duty Cycle Error	−0.4% ∼ 0.6% @262 MHz −1.6% ∼ 0.8% @442 MHz −1.4% ∼ 0.4% @1.02 GHz	−0.9% ∼ 0.4% @250 MHz −0.6% ∼ 0.9% @500 MHz −1.0% ∼ 1.0% @1 GHz	±0.6%	−1.5% ∼ 1.4% @1 GHz	±1.8%	±1% @400MHz ±3.5 @1 GHz	±1%	2% @1.25 GHz	±2%
Duty Cycle Corrector Resolution	3.5 ps	1.75 ps	120 ps	78.1 ps*	17.75 ps**	78.1***	N/A	N/A	N/A
Align with input clock	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Power consumption	1.96 mW @262 MHz 2.68 mW @442 MHz 6.5 mW @1.02 GHz	1.52 mW @250 MHz 3.03 mW @500 MHz 5.83 mW @1 GHz	20 mW @500 MHz	15 mW @1 GHz	43 mW @1.5 GHz	1.76 mW @400 MHz 3.6 mW @2 GHz	4.8 mW @1.3 GHz	150 mW @1.25 GHz	1.4 mW @1 GHz
p-p Jitter	23.64 ps @1.02 GHz	N/A	16.7 ps @500 MHz	12.9 ps @1 GHz	7 ps @1.5 GHz	28.45 @1 GHz	13.2 ps	19.6 ps @1.25 GHz	N/A
Area(mm^2)	0.01	0.01	0.68	0.23	0.053	0.025	0.2068	0.141	0.01
Experimental Results Type	Measurement	Simulation	Measurement	Measurement	Measurement	Measurement	Measurement	Measurement	Measurement

*: 1250 ps/16 = 78.1 ps (@800 MHz)
**: τ = 2272.7 ps/16 = 142 ps (@440 MHz)
resolution is $\tau * 0.125 = 142 * 0.125 = 17.75$ ps
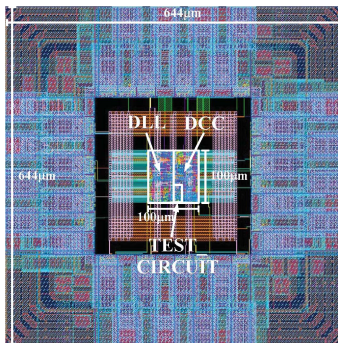***: 2500 ps/32 = 78.1 (@400 MHz)

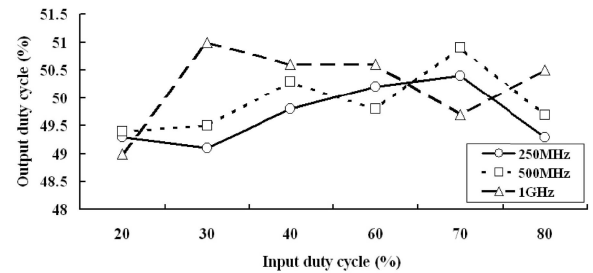Fig. 24.   Layout of the proposed HR-ADDCC.

Fig. 25.   Simulation results of the proposed HR-ADDCC.

lock-in time. In addition, the output clock of the PWCL is not phase aligned with the input clock [4], [7], [24]. Compared with prior works, the proposed ADDCC not only has a wider frequency range and a higher duty-cycle correction resolution but also has a wider input duty-cycle range and a phase alignment capability.

V. CONCLUSION

In this paper, phase-alignment and HR-ADDCCs were presented. The proposed PA-ADDCCs not only exhibited the phase alignment of the input and output clocks but also corrected the duty cycle of the output clock to 50% with a short locking time. The HR-ADDCC used a novel correction method without a half-cycle delay line, which can solve the delay mismatch problem in an advanced CMOS process with serious OCVs. In addition, the proposed ADDCC architectures can operate across a wide frequency range and achieved a wide range of input duty cycle. Thus, it is very suitable for duty-cycle correction applications, such as the DDR memory, I/O bus interface, and SoC applications.

ACKNOWLEDGMENT

REFERENCES

[1] F. Mu and C. Svensson, "Pulsewidth control loop in high-speed CMOS clock buffers," *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 134–141, Feb. 2000.

[2] P.-H. Yang and J.-S. Wang, "Low-voltage pulsewidth control loops for SOC applications," *IEEE J. Solid-State Circuits*, vol. 37, no. 10, pp. 1348–1351, Oct. 2002.

[3] S.-R. Han and S.-I. Liu, "A 500-MHz-1.25-GHz fast-locking pulsewidth control loop with presettable duty cycle," *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 463–468, Oct. 2004.

[4] K.-H. Cheng, C.-W. Su, and K.-F. Chang, "A high linearity, fast-locking pulsewidth control loop with digitally programmable duty cycle correction for wide range operation," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 399–413, Feb. 2008.

[5] Y.-M. Wang, C.-F. Hu, Y.-J. Chen, and J.-S. Wang, "An all-digital pulsewidth control loop," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jul. 2005, pp. 1258–1261.

[6] Y.-J. Wang, S.-K. Kao, and S.-I. Liu, "All-digital delay-locked loop/pulsewidth-control loop with adjustable duty cycles," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1262–1274, Jun. 2006.

[7] S.-R. Han and S.-I. Liu, "A single-path pulsewidth control loop with a built-in delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 40, no. 5, pp. 1130–1135, Jun. 2005.

[8] J. Gu, J. Wu, D. Gu, M. Zhang, and L. Shi, "All-digital wide range precharge logic 50% duty cycle corrector," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 760–764, Apr. 2012.

[9] S.-K. Kao and S.-I. Liu, "A wide-range all-digital duty cycle corrector with a period monitor," in *Proc. IEEE Conf. Electron Devices Solid-State Circuits*, Dec. 2007, pp. 349–352.

[10] Y.-M. Wang and J.-S. Wang, "An all-digital 50% duty-cycle corrector," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2004, pp. 925–928.

[11] B.-J. Chen, S.-K. Kao, and S.-I. Liu, "An all-digital duty cycle corrector," in *Proc. Int. Symp. VLSI Design, Autom.*, Apr. 2006, pp. 1–4.

[12] S.-K. Kao and S.-I. Liu, "All-digital fast-locked synchronous duty-cycle corrector," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 12, pp. 1363–1367, Dec. 2006.

[13] J.-H. Bae, J.-H. Seo, H.-S. Yeo, J.-W. Kim, J.-Y. Sim, and H.-J. Park, "An All-digital 90-degree phase-shift DLL with loop-embedded DCC for 1.6Gbps DDR Interface," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2007, pp. 373–376.

[14] J.-S. Wang, C.-Y. Cheng, J.-C. Liu, Y.-C. Liu, and Y.-M. Wang, "A duty-cycle-distortion-tolerant half-delay-line low-power fast-lock-in all-digital delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 45, no. 5, pp. 1036–1047, May 2010.

[15] B.-G. Kim, K.-I. Oh, L.-S. Kim, and D.-W. Lee, "A 500MHz DLL with second order duty cycle corrector for low jitter," in *Proc. IEEE Custom Integr. Circuits Conf.*, Jan. 2006, pp. 325–328.

[16] D. Shin, C. Kim, J. Song, and H. Chae, "A 7 ps Jitter 0.053 mm2 fast lock all-digital dll with a wide range and high resolution DCC," *IEEE J. Solid-State Circuits*, vol. 44, no. 9, pp. 2437–2451, Aug. 2009.

[17] J.-W. Ke, S.-Y. Huang, and D.-M. Kwai, "A high-resolution all-digital duty-cycle corrector with a new pulse-width detector," in *Proc. IEEE Conf. Electron Devices Solid-State Circuits*, Dec. 2010, pp. 1–4.

[18] D. Shin, W.-J. Yun, H.-W. Lee, Y.-J. Choi, S. Kim, and C. Kim, "A 0.17-1.4GHz low-jitter all digital DLL with TDC-based DCC using pulse width detection scheme," in *Proc. Eur. Solid-State Circuits Conf.*, Sep. 2008, pp. 82–85.

[19] S. Henzler, S. Koeppe, D. Lorenz, W. Kamp, R. Kuenemund, and D. Schmitt-Landsiedel, "A local passive time interpolation concept for variation-tolerant high-resolution time-to-digital conversion," *IEEE J. Solid-State Circuits*, vol. 43, no. 7, pp. 1666–1676, Jul. 2008.

[20] L. Vercesi, A. Liscidini, and R. Castello, "Two-Dimensions vernier time-to-digital converter," *IEEE J. Solid-State Circuits*, vol. 45, no. 8, pp. 1504–1512, Aug. 2010.

[21] J. Yu, F. F. Dai, and R. C. Jaeger, "A 12-bit vernier ring time-to-digital converter in 0.13 m CMOS technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 830–842, Apr. 2010.

[22] H.-J. Hsu, C.-C. Tu, and S.-Y. Huang, "A high-resolution all-digital phase-lock loop with its application to built-in speed grading for memory," in *Proc. IEEE Symp. VLSI Design Autom.*, Apr. 2008, pp. 267–270.

[23] D. Sheng, C.-C. Chung, and C.-Y. Lee, "An ultra-low-power and portable digitally controlled oscillator for SoC applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 11, pp. 954–958, Nov. 2007.

[24] R. Mehta, S. Seth, S. Shashidharan, B. Chattopadhyay, and S. Chakravarty, "A programmable, multi-GHz, wide-range duty cycle correction circuit in 45nm CMOS process," in *Proc. Eur. Solid-Stage Circuits Conf.*, Sep. 2012, pp. 257–260.

Ching-Che Chung (S'01–M'03) received the B.S. and Ph.D. degrees in electronics engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1997 and 2003, respectively.

He was a Post-Doctoral Researcher with National Chiao-Tung University from 2004 to 2008, working in the area of system-on-chip design methodologies and high-speed interface circuit design. In 2008, he joined the Faculty of the Computer Science and Information Engineering Department, National Chung Cheng University, Chia-Yi, Taiwan, where he is currently an Associate Professor. His current research interests include wireless and wireline communication systems, low-power and system-on-a-chip design technology, mixed-signal IC design and sensor circuits design, all-digital phase-locked loop, and all-digital delay-locked loop and its applications.

Duo Sheng (S'07–M'12) received the B.S. and M.S. degrees in electrical engineering from National Chung Cheng University, Chia-Yi, Taiwan, in 1997 and 1999, respectively, and the Ph.D. degree in electronics engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 2010.

He was with the Macronix Group, Hsinchu, from 1999 to 2009, engaged in system-on-a-chip (SoC) design, high-performance clocking IP development, and high-speed interface circuit design. He joined the Faculty of the Department of Electrical Engineering, Fu Jen Catholic University, Taipei, Taiwan, in 2010, where he is currently an Assistant Professor. His current research interests include low-power and high-speed digital integrated circuits and systems, all-digital clocking generator, and low-power SoC for biomedical applications.

Sung-En Shen received the M.S. degree in computer science and information engineering from National Chung Cheng University, Chia-Yi, Taiwan, in 2011.

He is currently a Design Engineer with the Logic IC Design Department, Etron Technology Inc., Hsinchu, Taiwan, working on USB 3.0 front-tend circuit design. His current research interests include system-on-a-chip design methodologies and all-digital duty-cycle corrector.