

An All-Digital Phase-Locked Loop for High-Speed Clock Generation

Ching-Che Chung and Chen-Yi Lee

Abstract—An all-digital phase-locked loop (ADPLL) for high-speed clock generation is presented in this brief. The proposed ADPLL architecture uses both a digital control mechanism and a ring oscillator and, hence, can be implemented with standard cells. The ADPLL implemented in a 0.3- μm one-poly-four-metal CMOS process can operate from 45 to 510 MHz and achieve worst case frequency acquisition in 46 reference clock cycles. The power dissipation of the ADPLL is 100 mW (at 500 MHz) with a 3.3-V power supply. From chip measurement results, the P_k - P_k jitter of the output clock is < 70 ps, and the root-mean-square jitter of the output clock is < 22 ps. A systematic way to design the ADPLL with the specified standard cell library is also presented in this brief. The proposed ADPLL can easily be ported to different processes in a short time. Thus, it can reduce the design time and design complexity of the ADPLL, making it very suitable for system-on-chip applications.

Index Terms—All-digital phase-locked loop (ADPLL), clock generator, frequency synthesizer, HDL, low jitter.

I. INTRODUCTION

PHASE-LOCKED loops (PLLs) are widely used for frequency synthesis applications [1], [5], [7]–[9]. For portable or mobile applications, lock-in time is very important since the PLL must support fast entry and exit from power management techniques [9]. Jitter less than $\pm 4\%$ of the clock cycle time is typically needed to avoid functional failures in a microprocessor [1]. Thus, how to design a fast-locking PLL with low-jitter performance in a short period is the goal for this research.

Due to high integration of very-large-scale integration (VLSI) systems, PLLs often operate in a very noisy environment. The digital switching noise coupled through power supply and substrate induces considerable noise into noise-sensitive analog circuits [1], [3]–[7]. Many analog approaches are proposed to improve the jitter performance of PLLs, such as choosing a narrow bandwidth or using a low-gain voltage-controlled oscillator (VCO) [5]. However, those analog approaches often result in long lock-in time and increasing design complexity of the PLLs.

Recently, all-digital phase-locked loops (ADPLLs) have become more attractive because they yield better testability, programmability, stability, and portability over different processes [2], [8], [9], and they can reduce the system turnaround time. The digitally controlled PLL (DCPLL) [2] has been proposed to achieve fast lock-in time, but due to the low sensitivity of the frequency detector and the resolution limitation of the digital-to-analog (D/A) converter, its jitter performance

Manuscript received February 4, 2002; revised August 26, 2002. This work was supported by the National Science Council of Taiwan, R.O.C., under Grant NSC90-2215-E-009-105.

The authors are with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: wildwolf@si2lab.org).

Digital Object Identifier 10.1109/JSSC.2002.807398

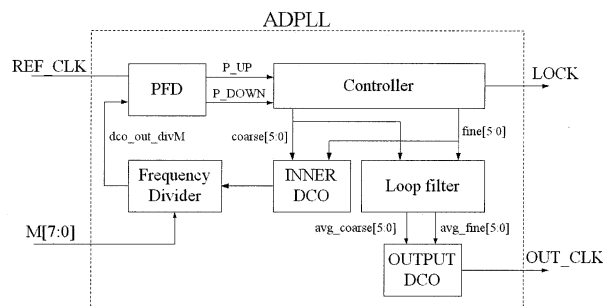


Fig. 1. Proposed ADPLL block diagram.

is worse than analog designs. The ADPLL proposed in [9] can achieve fine resolution and fast lock-in time; however, its digitally controlled oscillator (DCO) needs to be fully custom designed, making it difficult for porting to different processes as design requests. A complete cell-based ADPLL is proposed in [8], where fine-search delay matrix architecture is developed to improve the DCO's resolution. Also, two DCOs are used to reduce output clock jitter effectively. However, the proposed fine-search delay matrix occupies a large silicon area and has high power consumption.

In this brief, an ADPLL with a novel fine-tuning delay cell is presented. It can reduce both cost and design time for building a high-resolution cell-based DCO. The proposed frequency tracking algorithm, which uses an adaptive search step, can achieve fast lock-in time. The proposed ADPLL has been verified on silicon using TSMC 0.35- μm one-poly-four-metal (1P4M) CMOS process standard cells with 3.3-V power supply. It has a frequency range of 45–510 MHz with DCO resolution better than 5 ps. The measured root-mean-square (rms) jitter is < 22 ps and P_k - P_k jitter is < 70 ps over the frequency range of ADPLL. Power dissipation is < 100 mW at 45–510 MHz. Moreover, a systematic design approach that uses the advantages of digital VLSI is proposed in this brief for a truly portable and cost-effective ADPLL-based frequency synthesizer solution.

This brief is arranged as follows. Section II describes the proposed ADPLL architecture. The implementation of the proposed ADPLL using TSMC 0.35- μm 1P4M CMOS process is presented in Section III. Section IV shows simulation and measurement results of the prototype chip. Section V concludes with a summary.

II. PROPOSED ADPLL ARCHITECTURE

Fig. 1 shows the block diagram of the proposed ADPLL. There are four major building blocks in the proposed ADPLL, namely the phase/frequency detector (PFD), the DCO, the ADPLL controller, and the digital loop filter. Note that there are

two DCOs in the ADPLL; one is used for tracking the reference clock and the other is used for generating the output clock.

The output clock of INNER DCO is divided by M then connected to the PFD. The PFD detects the frequency difference and phase error between reference clock (REF_CLK) and divided output clock (dco_out_divM). It then generates an up (P_UP) signal or down (P_DOWN) signal to indicate that the INNER DCO should be speeded up or slowed down respectively. Then the ADPLL controller updates the INNER DCOs control code: coarse [5:0] and fine [5:0] to adjust the output frequency of INNER DCO. This INNER DCOs control code is also sent to the loop filter.

In ADPLL controller, an adaptive search step is used when it searches for the target frequency. The frequency acquisition starts from the middle frequency band of the DCO. If DCO can provide n different frequencies, the search step is $n/4$ in the initial state. When the output frequency is lower than the target frequency, the ADPLL controller adds a current search step to the DCO control code, and this increases the output frequency of the DCO. Oppositely, when the output frequency is higher than the target frequency, the ADPLL controller subtracts the DCO control code to lower the output frequency of the DCO. Whenever the PFDs output changes from P_UP to P_DOWN or *vice versa*, the search step is divided by two. After the search step reduces to 1 (i.e., one fine-tuning step of the INNER DCO), the frequency acquisition is done. Then the ADPLL controller enters phase-acquisition and phase-maintaining mode. In this mode, the ADPLL controller adjusts the fine-tuning control code of the INNER DCO to eliminate the phase error between REF_CLK and dco_out_divM whenever it receives the P_UP or P_DOWN from PFD.

The ADPLL's closed-loop response time is determined by the response time of the DCO, the delay time of the ADPLL controller, and the frequency divider. Therefore, the DCO's control code can only be updated at every m cycles, instead of every reference clock cycle. Here, m is determined by the closed-loop's response time. The worst case lock-in time for this frequency acquisition algorithm, in terms of reference clock cycles, is $T(n) = m * (1 + 2 * \log_2(n/2)) = m * (2 * \log_2(n) - 1)$.

After the ADPLL has finished frequency acquisition and phase acquisition, the INNER DCO's control code becomes converged to a fine-tuning range. However, the control code has small variations due to the following factors: the PFD's dead zone, the DCO's finite resolution, and input jitter. To further improve the jitter performance of the APDLL, the loop filter is used to filter out the resultant noise into the OUTPUT DCO. Thus, the loop filter detects the maximum INNER DCO control code and minimum INNER DCO control code within k reference clock cycles and then outputs $(\text{DCO control code}_{\max} + \text{DCO control code}_{\min})/2$ as the average DCO control code for the OUTPUT DCO.

III. CIRCUIT DESIGN

A. DCO

The proposed cell-based DCO is shown in Fig. 2(a). The DCO is implemented with TSMC 0.35- μm 1P4M CMOS standard cell library. It is separated into two stages: a coarse-tuning

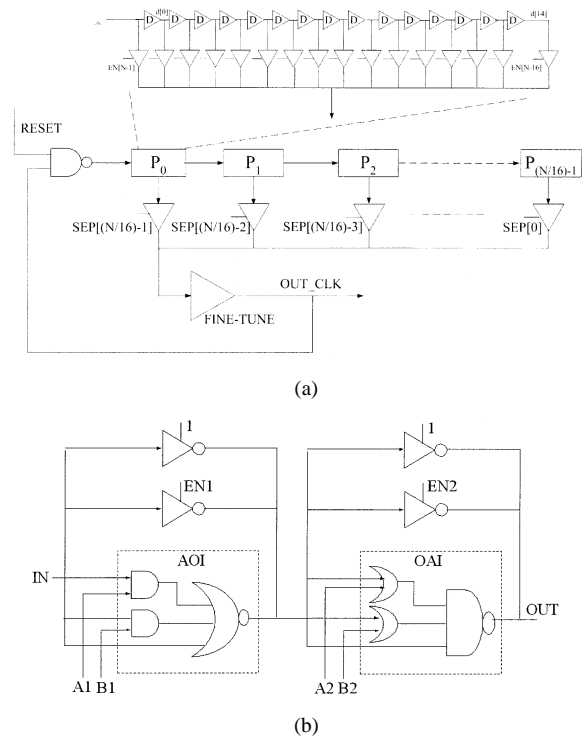


Fig. 2. (a) Architecture of the proposed DCO. (b) Fine-tuning delay cell.

stage and a fine-tuning stage. In the coarse-tuning stage, the coarse-tuning delay chain with 64-to-1 select-path architecture is used to provide different delays for coarse tuning. The 64-to-1 select-path architecture is implemented with tristate buffers. The DCO coarse-tuning encoder will encode $6(=\log_2(64))$ bits coarse-tuning control code into 64-paths selection control signals. This architecture has the advantage of minimum intrinsic delay time in the path selector to improve operating frequency of the DCO, and it can be easily modified to meet different specifications for different applications. To avoid a large loading capacitance appearing in the path selector's output, the path selector is partitioned into two stages. In the first stage, every 16 coarse-tuning delay blocks will select a partial output, and the second-stage path selector will select the final output. The delay time difference between two neighbor paths is determined by one coarse-tuning delay cell. The $T_{PHL} + T_{PLH}$ of one coarse-tuning delay cell is about 300 ps. Thus, when the DCO's coarse-tuning control code increases by one or decreases by one, the amount of the output clock's period will be changed by ± 300 ps.

To increase the frequency resolution of the DCO, a fine-tuning delay cell is added after the coarse-tuning stage. The schematic of the fine-tuning delay cell is shown in Fig. 2(b) [10]. The fine-tuning delay cell consists of an AND-OR-INV (AOI) cell and an OR-AND-INV (OAI) cell. Both the AOI cell and the OAI cell are shunted with two tristate buffers. Shunted tristate buffers can increase the controllable range of the fine-tuning delay cell. The controllable range of the fine-tuning delay cell should cover one coarse-tuning step (i.e., 300 ps). In the fine-tuning delay cell, in total 6 bits (EN1, A1, B1, EN2, A2, B2) can be controlled. Thus, in total $64(=2^6)$ different delays can be used. After HSPICE simulation, the lookup table

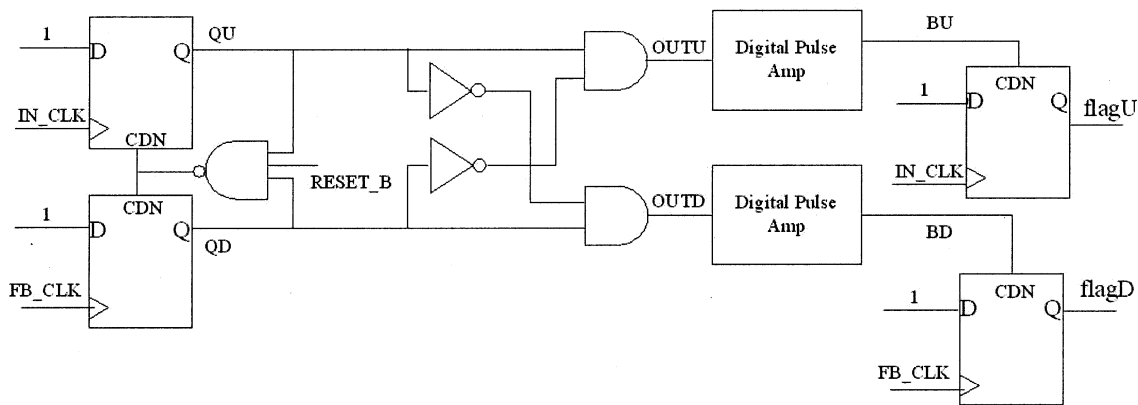


Fig. 3. PFD.

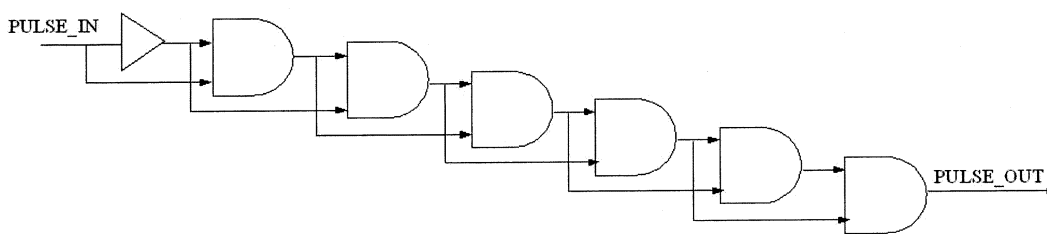


Fig. 4. Digital pulse amplifier.

for mapping the fine-tuning control code can be created. The DCO resolution can be improved to about 5 ps by adding a fine-tuning delay cell. The maximum output frequency of the DCO is 545 MHz (1.833 ns) and the minimum output frequency of the DCO is 41 MHz (24.261 ns) by HSPICE simulation.

B. PFD

The schematic of the proposed PFD is shown in Fig. 3. When the divided output clock (FB_CLK) leads the reference clock (IN_CLK), flagD generates a low pulse and flagU remains high. Oppositely, when FB_CLK lags IN_CLK, flagU generates a low pulse and flagD remains high. The ADPLL controller will be triggered by those signals. Since the ADPLL controller only needs to know that the FB_CLK leads IN_CLK or FB_CLK lags IN_CLK, the digital pulse amplifier can effectively minimize the dead zone of the PFD.

The schematic of the digital pulse amplifier is shown in Fig. 4. It uses the cascaded two-input AND architecture to increase the pulsewidth of OUTU and OUTD. The digital pulse amplifier enlarges the phase error between IN_CLK and FB_CLK, thus, the following D-flip-flops can detect it. When the phase error is less than ± 50 ps, both flagU and flagD will remain in high, and no trigger signal is sent to the ADPLL controller.

C. ADPLL Controller/Frequency Divider/Loop Filter

Hardware Description Language (HDL) is used to describe the ADPLL controller, the frequency divider, and the loop filter. We use a logic synthesizer to synthesize those modules to gate-level circuits with TSMC 0.35- μ m 1P4M CMOS cell library.

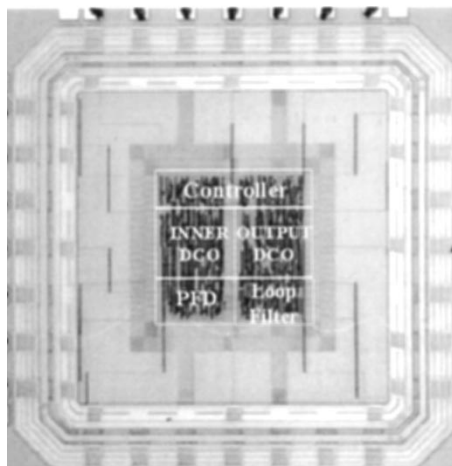


Fig. 5. Microphotograph of the ADPLL.

D. Systematic Design Approach

From the above description, a systematic way is provided to design the ADPLL with the specified standard cell library. First, HSPICE simulation of the fine-tuning delay cell should be performed and a lookup table for mapping the fine-tuning control code can be created after transistor-level circuit simulation. When the controllable range of the fine-tuning cell is determined, the suitable coarse-tuning cell, whose $T_{PHL} + T_{PLH}$ is less than or equal to the controllable range of the fine-tuning delay cell, can be selected from the cell library. The specifications of the output clock frequency determine the number of select paths in the coarse-tuning stage. The other modules of the ADPLL can be synthesized to the gate-level circuit by the logic

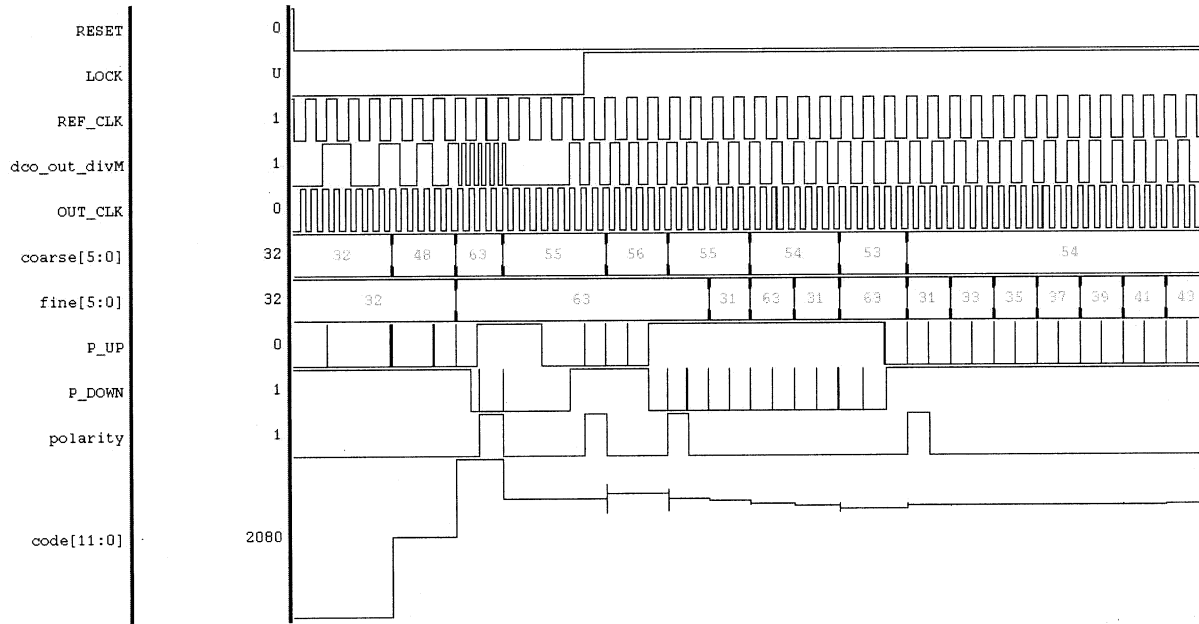


Fig. 6. Transient response of the ADPLL (at 200 MHz).

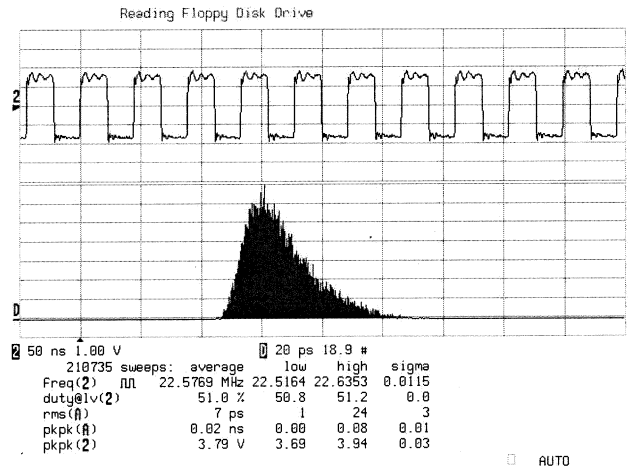
synthesizer. Thus, design time and complexity of design for the ADPLL can be reduced, and the proposed ADPLL architecture can easily be ported to different processes in a short time.

Fig. 5 shows the microphotograph of the ADPLL chip. We use auto placement and routing (APR) tools to generate the layout of the ADPLL. Since different interconnection delays may result in mismatches between INNER DCO and OUTPUT DCO, we use APR tools to generate one DCO layout first and then duplicate this DCO layout in the final APR process. Both the DCO and the PFD should have a maximum occupied area constraint to minimize the wire-loading effects during APR. The gate count of the ADPLL is 4800. The core size of the ADPLL is $840 \mu\text{m} \times 840 \mu\text{m}$ and the chip size including I/O pads is $2010 \mu\text{m} \times 2010 \mu\text{m}$.

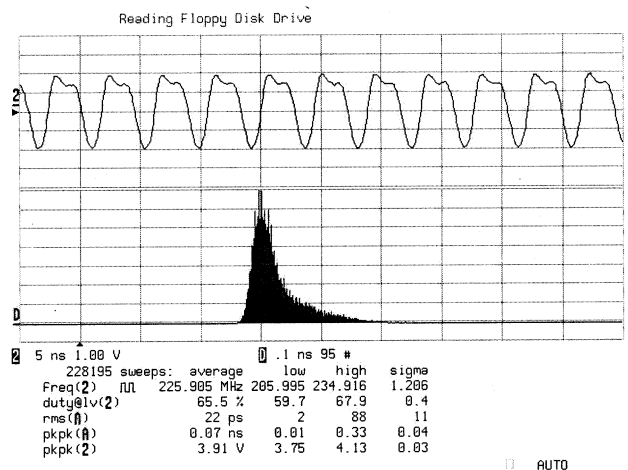
IV. EXPERIMENTAL RESULTS

Fig. 6 shows the transient response of the ADPLL, where the reference clock is 5 MHz, and the division ratio M is 40. Thus the output frequency is 200 MHz ($=5 \text{ MHz} * 40$). The code [11 : 0], which means {coarse [5 : 0], fine [5 : 0]}, is converged to a fine-tuning range of the INNER DCO in a short time. By using an adaptive search step in frequency acquisition, the ADPLL can finish frequency acquisition in $46(=2 * (2 * \log_2(2^{12}) - 1))$ reference clock cycles, where $m = 2$ in this worst case.

Due to the speed limitation of the I/O pad, the output frequency of the ADPLL must be lowered for testing. Fig. 7 shows the measured output waveform of the ADPLL with noisy digital circuitry ($\approx 600 \text{ mVpp}$ supply noise) at 45 and 450 MHz, respectively. The signal at Channel 2 shows the OUT_CLK signal divided by two and the signal at Channel D shows the long-term $P_k - P_k$ jitter histogram over 200 000 sweeps. We use LeCory LC584A to measure the PLL's output signal. The rms jitter and $P_k - P_k$ jitter at 45 MHz is 7 and 20 ps, respectively. The rms jitter and $P_k - P_k$ jitter at 450 MHz is 22 and 70 ps, respectively.



(a)



(b)

Fig. 7. Jitter histogram of the ADPLL at (a) 45 MHz and (b) 450 MHz.

TABLE I
PERFORMANCE COMPARISONS

Performance Parameter	Proposed ADPLL	[1]	[2]	[8]	[9]
Process	0.35 μ m CMOS	0.25 μ m CMOS	0.60 μ m CMOS	0.6 μ m CMOS	0.5 μ m CMOS
Area	0.71mm ²	0.09mm ²	0.83mm ²	2.75mm ²	0.71 mm ²
Approach	All-Digital Cell-Based	Analog	Semi-Digital	All-Digital Cell-Based	All-Digital
Power dissipation	100mW (@500MHz)	25mW	105mW (@400MHz)	315mW @ (800MHz)	39.6mW @ (100MHz)
Max. Lock time	< 46 cycles	< 720 cycles	< 16 cycles	< 25 cycles	< 50 cycles
Min. Frequency	45MHz	8.5MHz	300MHz	360MHz	50MHz
Max. Frequency	510MHz	660MHz	800MHz	800MHz	550MHz
Supply voltage	3.3V	1.9V	3.3V	3.3V	3.3V
Output jitter (P_k - P_k)	70ps	80ps	149ps	60ps	125ps

Table I lists the comparisons among different PLLs. The proposed ADPLL has shorter lock-in time and better jitter performance than the analog PLL [1] and the ADPLL [9]. Although the DCPLL [2] can achieve fast locking, its jitter performance is worse than the proposed design. The proposed ADPLL has smaller area and lower power consumption than the cell-based ADPLL [8]. The proposed ADPLL can achieve fast locking in 46 reference clock cycles, and it has better portability than the other designs in Table I.

V. CONCLUSION

In this paper, an ADPLL is presented. The ADPLL can be implemented with standard cells, and it has good portability over different processes. The ADPLL is implemented in a 0.35- μ m 1P4M CMOS standard cell library and can operate from 45 to 510 MHz. The P_k - P_k jitter of the output clock is less than 70 ps, and the rms jitter of the output clock is less than 22 ps. A systematic way to design the ADPLL is also presented in this brief. The proposed ADPLL can reduce design time and circuit complexity. Therefore, it is very suitable for system-on-chip applications.

ACKNOWLEDGMENT

The authors would like to thank their colleagues within the SI2 group of National Chiao Tung University for many fruitful discussions. The multiproject chip support from the National Science Council of Taiwan/Chip Implementation Center is also acknowledged.

REFERENCES

- [1] H.-T. Ahn and D. J. Allstot, "A low-jitter 1.9-V CMOS PLL for ultra-SPARC microprocessor applications," *IEEE J. Solid-State Circuits*, vol. 35, pp. 450–454, May 1999.
- [2] I. Hwang, S. Lee, S. Lee, and S. Kim, "A digitally controlled phase-locked loop with fast locking scheme for clock synthesis application," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2000, pp. 168–169.
- [3] K. Minami, M. Fukaishi, M. Mizuno, H. Onishi, K. Noda, K. Imai, T. Horiuchi, H. Yamaguchi, T. Sato, K. Nakamura, and M. Yamashina, "A 0.1- μ m CMOS 1.2-V 2-GHz phase-locked loop with gain compensation VCO," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, May 2001, pp. 213–216.
- [4] A. Maxim, B. Scott, E. M. Schneider, M. L. Hagge, S. Chacko, and D. Stuurca, "A low-jitter 125–1250-MHz process-independent and ripple-poleless 0.18- μ m CMOS PLL based on a sample-reset loop filter," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1673–1683, Feb. 2001.
- [5] I. A. Young, M. F. Mar, and B. Bhushan, "A 0.35- μ m CMOS 3-880-MHz PLL N/2 clock multiplier and distribution network with low jitter for microprocessors," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 1997, pp. 330–331.
- [6] C.-H. Lee, J. Cornish, K. McClellan, and J. Choma, Jr., "Design of low-jitter PLL for clock generator with supply noise insensitive VCO," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, June 1998, pp. 233–236.
- [7] D. W. Boerstler, "A low-jitter PLL clock generator for microprocessors with lock range of 340–612 MHz," *IEEE J. Solid-State Circuits*, vol. 34, pp. 513–519, Apr. 1999.
- [8] T.-Y. Hsu, C.-C. Wang, and C.-Y. Lee, "Design and analysis of a portable high-speed clock generator," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 367–375, Apr. 2001.
- [9] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, "An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors," *IEEE J. Solid-State Circuits*, vol. 30, pp. 412–422, Apr. 1995.
- [10] J.-J. Jong and C.-Y. Lee, "A novel structure for portable digitally controlled oscillator," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, May 2001, pp. 272–275.