A Low-Noise Fast-Lock Phase-Locked Loop with Adaptive Bandwidth Control

Joonsuk Lee, *Student Member, IEEE,* and Beomsup Kim, *Senior Member, IEEE*

***Abstract*—This paper presents a salient analog phase-locked loop (PLL) that adaptively controls the loop bandwidth according to the locking status and the phase error amount. When the phase error is large, such as in the locking mode, the PLL increases the loop bandwidth and achieves fast locking. On the other hand, when the phase error is small, this PLL decreases the loop bandwidth and minimizes output jitters. Based on an analog recursive bandwidth control algorithm, the PLL achieves the phase and frequency lock in less than 30 clock cycles without pre-training, and maintains the cycle-to-cycle jitter within 20 ps (peak-to-peak) in the tracking mode. A feed forward-type duty-cycle corrector is designed to keep the 50% duty cycle ratio over all operating frequency range.**

***Index Terms*—Adaptive bandwidth PLL, analog implementation, clock recovery, fast locking time, frequency hopping, gear-shifting algorithm, low jitter, phase-locked loops, time-varying channel.**

I. Introduction

PHASE-LOCKED loops (PLL's) have been widely used in high-speed data communication systems such as Ethernet receivers, disk drive read/write channels, digital mobile receivers, high-speed memory interfaces, and so forth, because PLL's efficiently perform clock recovery or clock generation with relatively low cost. Those PLL's used in the systems are required to generate low-noise or low-jitter clock signals and at the same time need to achieve fast locking.

Conventional analog PLL's in clock recovery applications use a narrow-band loop filter to reduce output jitters at the expense of elongated locking time. In order to improve the locking-time characteristics, digital or hybrid analog/digital PLL's with a loop bandwidth stepping capability have been studied [1], [2]. Since the stepping hardware is implemented with complex digital building blocks, these PLL's usually suffer from high power dissipation, low operating speed and large die size. In order to reduce consuming power and die size, simpler algorithms such as a gear-shifting or a lock-detection algorithm were attempted [3], [4]. The PLL's with such algorithms control the loop bandwidth according to a prestored charge-pump current control sequence in memory during the start-up mode. However, in clock recovery applications such as HDD and DVD, where the channel characteristics vary in time, the prestored control sequence cannot make the PLL's respond properly to unpredictable phase fluctuation, instant frequency shift, and time-varying jitter because the sequence was calculated with preknown fixed noise statistics.

Discrete-time PLL's, which are programmed on DSP processors, based on a recursive least squared (RLS) algorithm [5] or the Kalman filter algorithm [6] can respond to such unpredictable jitter variations, but require enormous amount of hardware. The outputs generated from the discrete-time PLL's are in a digital domain, and therefore the discrete-time PLL's require digital-to-analog converters (DAC) and an analog-to-digital converter (ADC) to sample input signals for detection. Slow signal-processing speed of the digital-to-analog conversion in the discrete-time PLL's limits the operating frequency and confines the use of the PLL's to the applications dealing with low-frequency signals like digital wireless base stations.

This paper presents a new analog adaptive PLL (AAPLL) architecture capable of varying the loop bandwidth according to an adaptively updated control sequence under a time-varying noise environment. Since the control sequence is generated from analog signal processing, the PLL operates at several hundred megahertz and can be easily modified to run at gigahertz frequency ranges.

This paper consists of five sections including the present section. Section II describes the AAPLL architecture and the analog adaptive bandwidth-control algorithm. Stability and jitter analysis for the AAPLL are given in Section III. AAPLL locking behaviors are also discussed in this section. Section IV shows the AAPLL IC implementation and measurement results. Finally, a brief summary of this paper is given in Section V.

II. Recursive Equation and Analog Loop Bandwidth Controller

In this section, a recursive bandwidth update algorithm for the analog adaptive controller and its implementation are described.

A. Adaptive Bandwidth Control

As mentioned in the introduction, a common approach to improve the locking speed of a PLL is to use a gear-shifting method for loop bandwidth control. In such a PLL, when fast locking is required, as in the initial frequency/phase acquisition mode, the loop bandwidth of the PLL is expanded by the increased charge-pump current or the phase detector gain [2]–[4]. Zero phase start (ZPS) is also helpful to reduce the phase acquisition time [4], but limited to the case when the initial-frequency locking has been already established. For the case where rapid initial-frequency locking is required, various techniques with a prestored gear-shifting sequence have been studied [4],

Manuscript received October 29, 1999; revised February 23, 2000.

J. Lee was with the Boston Design Center, IBM Microelectronics, Lowell, MA 01851 USA. He is now with the Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea.

B. Kim is with the Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea (e-mail: bkim@ee.kaist.ac.kr).

Publisher Item Identifier S 0018-9200(00)06435-0.

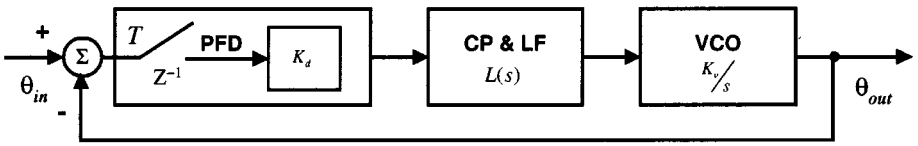


Fig. 1. Linearized model of a CP-PLL.

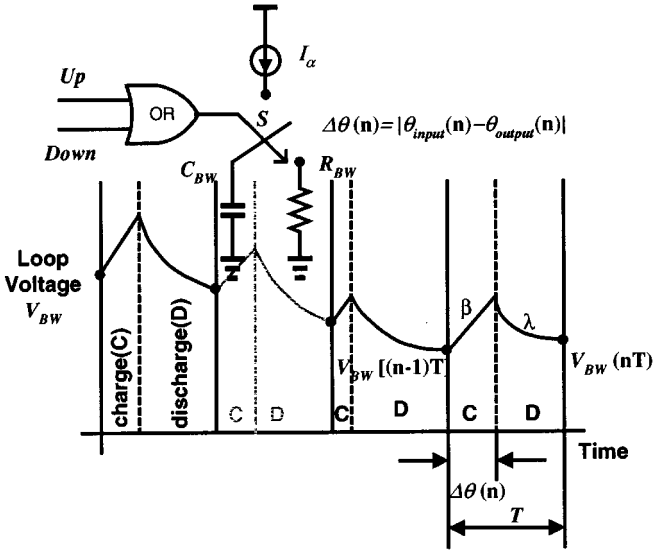


Fig. 2. Conceptual diagram of an analog adaptive controller.

[5]. However, in the case where the channel characteristics vary in time, such as in a disk drive, the prestored gear-shifting sequence is not helpful. Unpredictable phase fluctuation, instant frequency shift, and varying input jitter force such a PLL to use an indefinite wide loop bandwidth in order not to lose the locking. Although a discrete-time adaptive PLL can adjust the bandwidth according to the input noise statistics, it still requires complex hardware and its applications are limited to the low frequency operating systems [5].

A linearized model of a charge-pump PLL (CP-PLL) is shown in Fig. 1. The transfer function for z domain is represented by

$$H(z) = \frac{K_d K_v L'(z) z^{-1}}{1 + K_d K_v L'(z) z^{-1}} \tag{1}$$

where $L'(z) = T \cdot \mathbf{Z}\{\mathbf{L}^{-1}[L(s)/s]|_{t=nT}\}$. Here K_d and K_v are the phase detector gain given by $I_{\rm CP}/2\pi$ and the voltage-controlled oscillator (VCO) gain given by df/dv, respectively, and $L'(z)$ is the z-transform of the sampled version of $L(s)/s$, where $L(s)$ is the PLL loop filter in s domain given by $R_1 + 1/sC_1$ if a simple passive low-pass filter (R_1, C_1) is assumed to be used as a loop filter. The quantity $K_d K_v R_1 T$ is called the PLL loop gain $K_{\rm loop}$.

Discrete-time PLL's that have an adaptive stepping capability can control the loop bandwidth by a loop-gain update equation minimizing the RLS error [7]. However, it is difficult to fully implement the update equation used in the discrete-time PLL because it requires a significant amount of die size and power consumption. A simpler but still an effective loop gain update equation is required and used in the proposed AAPLL. The update equation is given by (2), in terms of the loop gain $K_{\rm loop}$ that is proportional to the loop bandwidth.[1]

$$K_{\rm loop}(n+1) = \lambda \cdot K_{\rm loop}(n) + \alpha \cdot |\theta_{\rm input}(n) - \theta_{\rm output}(n)|. \tag{2}$$

Here, λ is a forgetting factor that has a positive value close to but less than unity. α is a coefficient that normalizes and converts the absolute value of input–output phase errors from radians to dimensionless numbers. The loop gain $K_{\rm loop}(n+1)$ is calculated by a recursive manner according to (2). When the input–output phase error becomes zero, the forgetting factor λ makes the loop gain converge to zero as the discrete time nT increases. Equation (2) reflects the most recent input–output phase error $|\theta_{\rm input}(n) - \theta_{\rm output}(n)|$ most significantly. This recursive relation is similar to the RLS algorithms commonly used for an estimator [8].

The loop gain $K_{\rm loop}(n+1)$ at time $(n+1)T$ is calculated as the weighted sum of the present loop gain $K_{\rm loop}(n)$ and the absolute value of the present input–output phase error, $|\theta_{\rm input}(n) - \theta_{\rm output}(n)|$ at time nT. The equation indicates that the loop gain, thus the loop bandwidth, rapidly grows when the recent absolute phase errors become large, and greatly improves the PLL loop tracking capability. When the recent absolute phase errors become small, the loop gain shrinks, as does the loop bandwidth, because the first part of (2) dominates. The reduced loop bandwidth improves the PLL's input jitter rejection capability. Therefore, (2) satisfies the necessary loop bandwidth control under the presence of unpredictable jitter variation.

B. Analog Adaptive Bandwidth Control

Equation (2) is achieved by a CP-PLL with a small amount of extra hardware. The second term of (2), an absolute phase error, is obtained from outputs of a phase frequency detector (PFD). Since the PFD and the following charge-pump circuit generate *up/down* current signals proportional to the input–output phase difference, simply combining these *up/down* signals through an OR gate gives the absolute phase error signal $|\Delta\theta(n)| = |\theta_{\rm input}(n) - \theta_{\rm output}(n)|$ at time nT. Fig. 2 shows a conceptual diagram of how the recursive loop gain of the AAPLL is calculated in the controller. The bandwidth voltage $V_{\rm BW}$ becomes the bias voltage of the following current source in the charge-pump circuit and controls the amount of charge-pump currents. The current switch S steers the current proportional to the phase error and increases the voltage across the capacitor $C_{\rm BW}$ by the corresponding amount at a constant rate while the resistor $R_{\rm BW}$ exponentially discharges the capacitor. The resistor and the capacitor realize the first part of (2) with the forgetting factor λ. As

[1]Here, the loop bandwidth and the loop gain have the following relationship: $K_{\rm loop} = W_{\rm loop}/f$.

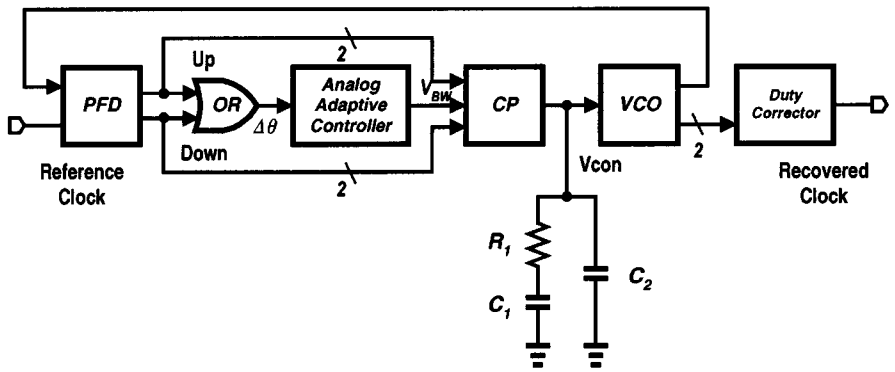


Fig. 3. AAPLL total block diagram.

derived in the Appendix, in the steady state the voltage across the capacitor $C_{\rm BW}$ at time $(n+1)T$ is given by

$$V_{\rm BW}[(n+1)T] = \lambda \cdot V_{\rm BW}[nT] + \beta \cdot |\Delta\theta(n)|. \qquad (3)$$

Here, the forgetting factor λ equals $\exp[-T/R_{\rm BW}C_{\rm BW}]$, β is $(I_\alpha/C_{\rm BW}) \cdot (T/2\pi) \cdot \lambda$, and I_α is the amount of the charging current in the controller.

The loop gain is asymptotically proportional to the bandwidth voltage $V_{\rm BW}$ governed by (3) because the charge-pump current is directly controlled by this voltage. It means that the bandwidth of the AAPLL follows (2).

III. Circuit Implementation

This section describes the circuit implementation of the adaptive bandwidth controller, the charge-pump circuit, the VCO, and the duty cycle correction circuit. Fig. 3 shows the overall block diagram of the AAPLL, which modifies a conventional PLL by attaching an analog adaptive bandwidth-controlling block. Due to the minor change, the AAPLL is easily applicable to various PLL applications and still takes advantage of the full adaptability.

A. Adaptive Bandwidth Controller and Charge Pump

The well-designed PFD is used as a phase-detecting block instead of a mixer, though the input signal frequency is high, in order to achieve a wideband capturing capability. The PFD shown in Fig. 4 consists of two simplified true single-phase clock (TSPC) D-flip–flops and one NOR gate. Since the input frequency of the AAPLL is selected to recover the clock signal in DVD systems, whose clock frequency is about 250 MHz, the PFD should generate *up* and *down* signals at such a high speed. In order to minimize the abnormal operation of the PFD, TSPC D-flip–flops are used as leaf cells since these intrinsic delays are smaller than those of conventional ones.

The adaptive bandwidth controller shown in Figs. 3 and 6 consists of an OR gate and a differential switch, which takes the differential signals from the OR gate. The OR gate sums the *up* and *down* signals generated from the PFD and gives the absolute phase error. The differential switch controls current paths from $V_{\rm dd}$ to the bandwidth capacitor $C_{\rm BW}$ according to the phase difference $|\theta_{\rm input}(n) - \theta_{\rm output}(n)|$ for one clock period. When the phase difference signal is mostly on over a period, such as in the initial-phase acquisition state, the charging rate

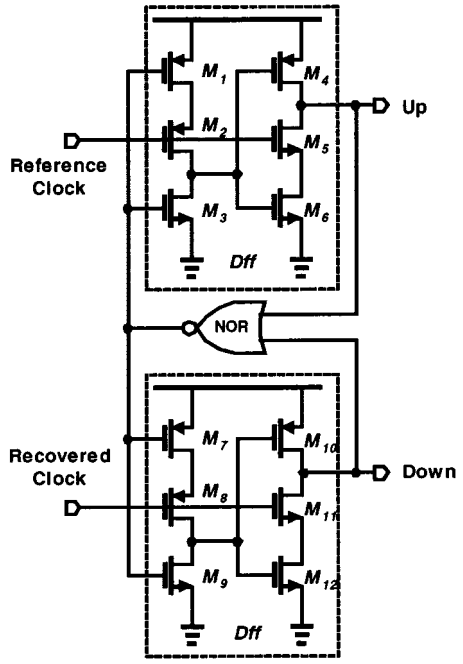


Fig. 4. PFD schematic with simplified TSPC D-flip–flops.

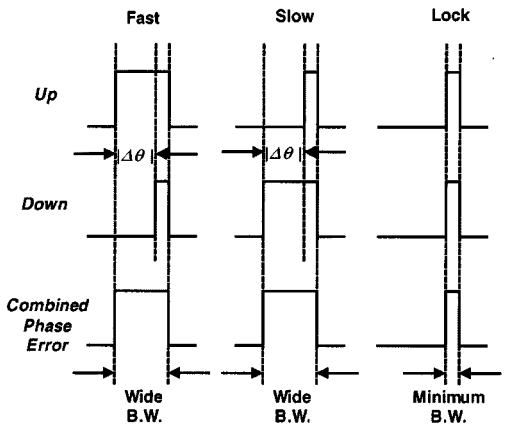


Fig. 5. *Up/Down* and phase-error signal diagram.

of the capacitor $C_{\rm BW}$ exceeds the discharging rate. Hence the capacitor voltage $V_{\rm BW}$ of the capacitor $C_{\rm BW}$ and the pumping current in the charge pump increase. As a result, the phase detector gain increases and so does the loop bandwidth. On the other hand, when the phase error signal is off for the most part

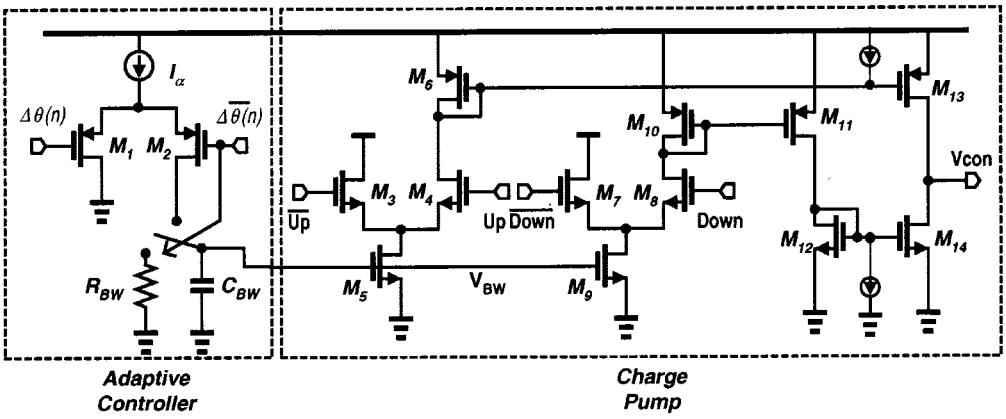


Fig. 6. Adaptive bandwidth controller and CP schematics.

of one period, such as in the tracking state, the discharging rate exceeds the charging rate and the capacitor voltage decreases. Therefore both the phase-detector gain and the loop bandwidth decrease. In the steady-state tracking mode, the AAPLL loop bandwidth can be very narrow because the phase error becomes zero. However, the AAPLL still maintains the minimum loop bandwidth even in such a case because of the *up/down* signals generated from the set/reset type PFD, as shown in Fig. 4. In the zero-phase-error and perfect locking case in Fig. 5, the OR-gated *effective* phase-error signal can still supply currents to the bandwidth capacitor. The statistical variation of the input signal also contributes to maintain this minimum bandwidth. Fig. 5 shows the relation between the phase error and the bandwidth control of the AAPLL.

Fig. 6 shows the circuit diagram of the analog adaptive bandwidth controller with a charge-pump circuit. As mentioned before, the voltage $V_{\rm BW}$ across the capacitor $C_{\rm BW}$ in parallel with a resistor $R_{\rm BW}$ controls the phase-detection gain K_d. In order to control the discharging rate of the capacitor $C_{\rm BW}$, a voltage-controlled resistor (VCR) $R_{\rm BW}$, as shown in Fig. 7, is used. The VCR is designed to have fully linear *I–V* characteristics for a given power supply range. By adjusting the magnitude of the bias current in the VCR branches, the resistance of $R_{\rm BW}$ is changed and so does the discharging rate of the capacitor $C_{\rm BW}$.

The output node of $C_{\rm BW}$ is connected to the gates of nMOS transistors, and controls the charge-pump current by adjusting the bias point of M_5 and M_9 in Fig. 6. The charge pump consists of two differential input stages, a mirror stage, an output stage, and two small extra current sources. These two small current sources help the rapid turn-on/off operation for MOS′, M_{13} and M_{14}. The differential PFD signals drive the charge-pump inputs. When the *down* signal goes high, the current controlled by the voltage of the bandwidth capacitor $C_{\rm BW}$ is drawn from the loop filter. When the *up* signal goes high, the same amount of current is supplied to the loop filter.

B. Voltage-Controlled Oscillator (VCO)

A four-stage VCO as shown in Fig. 8 is used for the AAPLL. The basic delay cell consists of six transistors. The cross-coupled pMOS transistors, M_3 and M_4, guarantee the differential operation of the delay cell without a tail-current bias. Auxiliary pMOS transistors, M_5 and M_6, control the oscillation frequency. Unlike

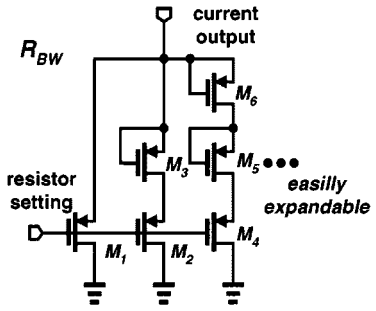


Fig. 7. VCR schematic.

conventional differential VCO's with a current bias, this VCO allows the AAPLL to operate under a single 1.5-V power supply, consuming 1.5 mA. Because the output signal of the VCO swings rail-to-rail, no additional level shifter with a carefully designed replica bias circuit is required to generate CMOS level outputs. The latch, configured with pMOS′, M_3 and M_4, sharpens the edge of the output signal so that the added noise has little chance to be converted as jitters. Eventually this latch helps the reduction process of the VCO jitter [9], [10].

C. Duty-Cycle Corrector

Maintaining a 50% duty-cycle ratio for a clock signal is extremely important in most high-speed clock recovery and clock generation applications because several systems, such as double-data rate (DDR) SDRAM's and pipelined microprocessors, use negative transition edges of a clock signal in order to increase total system throughputs. This is often achieved by a VCO running at twice as high as the desired clock frequency, and then dividing the VCO frequency by 2. Other approaches use a feedback-type duty-cycle corrector. Since precise placement of the falling edge between two successive rising edges of the VCO output signal is generally controlled by an additional feedback loop, the duty-cycle correctors require an extra training period to stabilize the feedback loop.

The AAPLL uses a feed forward-type duty-cycle corrector instead of the feedback type in order to eliminate the extra feedback hardware and the training period, as shown in Fig. 9(a). The duty cycle corrector utilizes multiphase signals generated from a multistage differential VCO. The signal $\bar{A}$ in Fig. 9(b) selected from the multiphase signals turns on MOS′, M_1 and

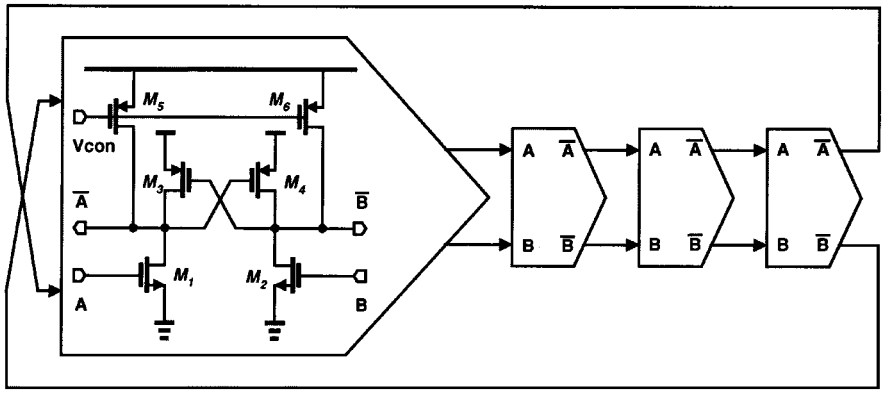
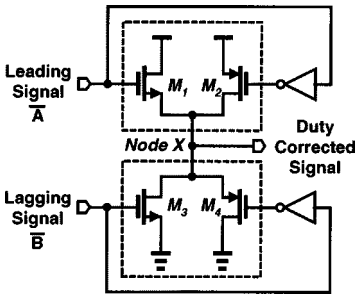
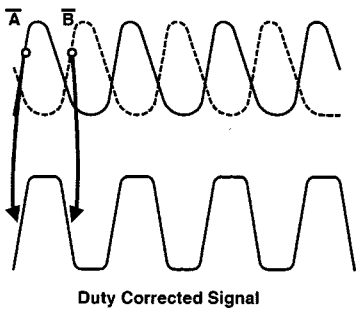


Fig. 8. Four-stage VCO.



(a)



(a)

Fig. 9. Feed forward-type duty-cycle corrector. (a) Duty-cycle corrector schematic. (b) Conceptual diagram of the correcting operation.

M_2, and charges the output node X of the duty-cycle corrector almost instantaneously, because the discharge path of the node X is already off due to the signal $\bar{B}$. The signal $\bar{B}$, which is also selected from the multiphase signals, is the one whose rising edge is shifted by 180° in phase from that of $\bar{A}$. Similarly, the signal $\bar{B}$ rapidly discharges the node X and delivers the desired 50% duty-cycle signal. Since this duty-cycle correction circuit consists of only two transmission gates and two inverters, the silicon area is minimal and the power consumption is negligible. In HSPICE simulation, the proposed duty-cycle corrector keeps the output duty cycle almost perfectly at 50% with the input duty cycles varying from 10 to 90%.

IV. Analysis and Simulation

In this section, the stability of the AAPLL is analyzed for the adaptively generated loop sequence, and behavioral simulation results for fast lock and large jitter reduction are described.

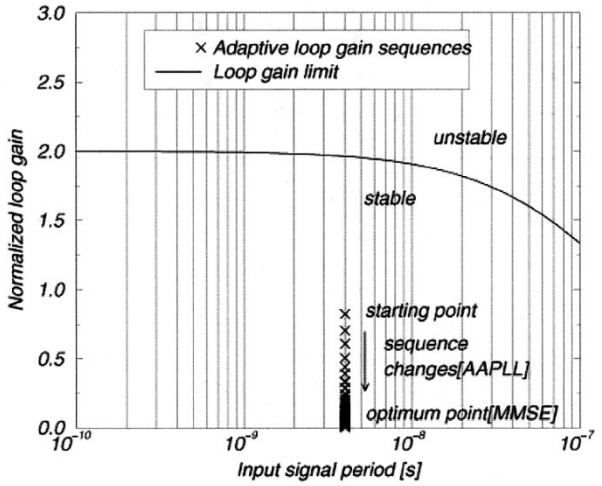


Fig. 10. Stability diagram for loop gain $K_{\rm loop}$.

A. Stability

Since the AAPLL automatically changes the loop bandwidth, a careful loop stability analysis is required. As mentioned in the previous section, an analog adaptive controller adjusts the phase-detector gain of the CP-PLL. Therefore, stability checking for the PLL for each different phase gain should be accomplished first. A complete stability analysis for the CP-PLL is cumbersome because a PLL operates in both a linear and a nonlinear region. A simplified stability analysis for a second-order CP-PLL [8] is used in this section. When the criterion is extended to include the logic delay effect, it can be expressed as

$$K_{\rm loop}[n] < \frac{2}{1+\frac{1}{2}\cdot\frac{T-\tau_d}{\tau}}. \tag{4}$$

Here, T, τ_d, and τ are the clock period, the logic delay, and the *RC* time constant of a loop filter respectively. The stability limit for the loop gain $K_{\rm loop}$ of the AAPLL is derived and simulated using this criterion as shown in Fig. 10. The adaptively generated loop gain sequence by the recursive equation is also shown in the same figure to verify the AAPLL stability. The sequence converges to the minimum bandwidth and the amplitude of this bandwidth is almost similar to that derived from the MMSE criterion [4]. Equation (4) can be written to obtain the stability criterion for the bandwidth voltage $V_{\rm BW}$ by solving a MOS I-V

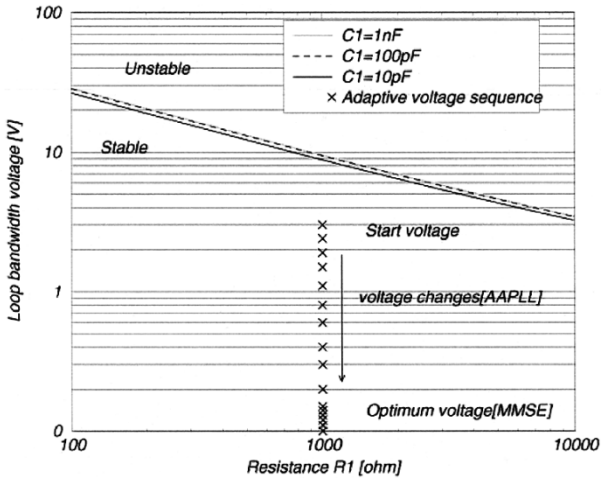


Fig. 11. Stability limit graph for bandwidth voltage $V_{\rm BW}$.

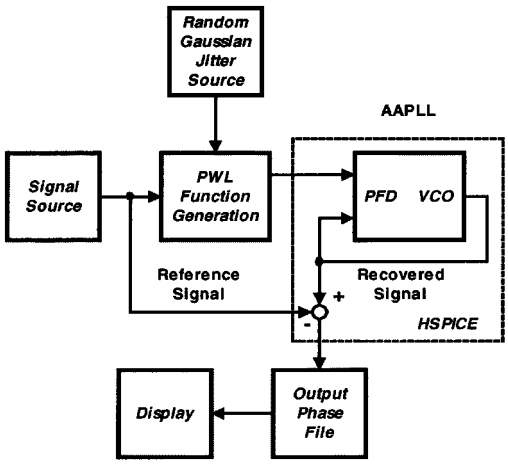


Fig. 12. Simulation setup for the locking behavior measurement.

characteristic equation for M_5, M_9 in Fig. 6 assumiing a saturation condition.

$$V_{\rm BW}[n] < \sqrt{\frac{4\pi}{\left(1+\frac{1}{2}\cdot\frac{T-\tau_d}{\tau}\right)\cdot(K_N K_V R_1 T)}} + V_T. \quad (5)$$

Here, K_N, K_V, V_T, and R_1 are $(\mu_N C_{\rm ox})/2\cdot(W/L)_{5,9}$, the VCO gain, the nMOS threshold voltage, and the size of a resistor of the loop filter in Fig. 3, respectively. Here, $(W/L)_{5,9}$ is the size of M_5, M_9. The stability limit for the bandwidth voltage, which is one of the observable values in the measurement setup, is visualized in Fig. 11 for various capacitor C_1 and resistor R_1 values. The figure shows that all the sequences $K_{\rm loop}[n]$ and $V_{\rm BW}[n]$ are within the stable region for the various resistor and capacitor values.

B. Output Jitter

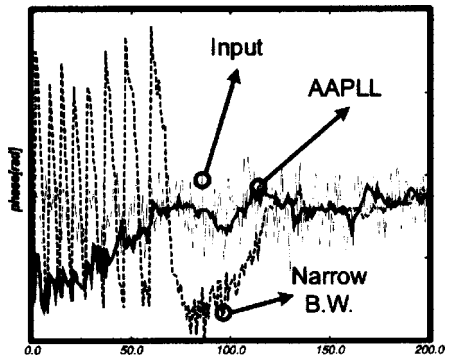
Recently, it was reported that a CP-PLL has an optimum loop bandwidth that generates minimum jitter in the steady state [11]. A clean tone, that is assumed to have only noise floor and no random walking phase noise, is used as a reference signal for the jitter derivation. Because the AAPLL eventually achieves the steady state locking with a clean reference signal like other conventional PLL's, the output cycle-to-cycle jitter of the AAPLL can be calculated by

$$\Delta\tau_{\rm rms} = \left(\delta\tau_{\rm rms1} + \frac{T^2}{2\pi}K_v\Delta V_{\rm rms4}\right)\cdot\sqrt{\frac{1}{2K_{\rm loop}}} + \left(\delta\tau_{\rm rms2} + \frac{T}{2\pi}\frac{\Delta I_{\rm rms3}}{K_d}\right)\cdot\sqrt{\frac{K_{\rm loop}}{2}}. \quad (6)$$

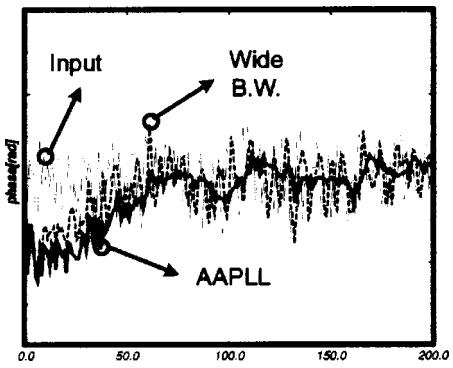
Here, $\delta\tau_{\rm rms1}$, $\delta\tau_{\rm rms2}$, $\Delta I_{\rm rms3}$, and $\Delta V_{\rm rms4}$ are the internal jitter from the VCO, the jitter of the input signal, the rms value of the charge-pump current variation, and the rms value of VCO control voltage noise in the steady state, respectively.

C. Behavioral Simulation of a Locking Feature

Closed-form analysis of locking behaviors for the AAPLL is difficult because of its nonlinear operation. In this paper, a simulation-based approach like the Monte Carlo Method is used instead. The AAPLL is modeled in a SPICE circuit simulator and extensively tested by the circuit simulator. Fig. 12 shows the simulation setup for the AAPLL. Fig. 13 compares the simulated locking behavior of the AAPLL with that of a conventional PLL. The bandwidths of the conventional PLL are selected to have two typical values. One is optimized for the initial locking, and the other for the steady-state tracking. The gray line in Fig. 13(a) indicates an incoming signal in the phase domain. The solid line in the figures shows the phase of the AAPLL output signal from initial locking to steady-state tracking. The phase variation of the conventional PLL optimized for steady-state tracking with a narrow bandwidth is shown in the same figure as a dashed line.



(a)



(b)

Fig. 13. Simulation results for the locking behavior of the AAPLL and conventional ones. (a) Fixed narrow-bandwidth PLL. (b) Fixed wide-bandwidth PLL.

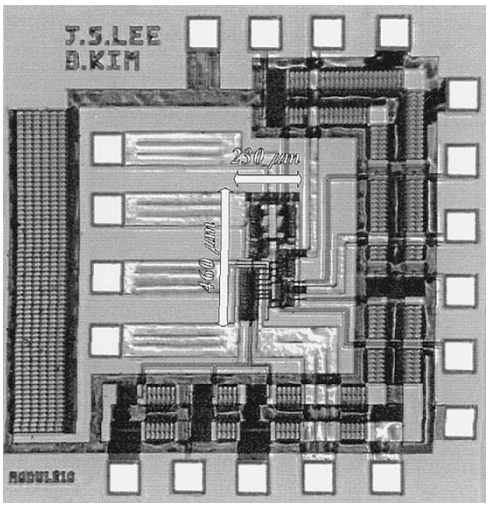


Fig. 14. Micrograph of the fabricated AAPLL chip.

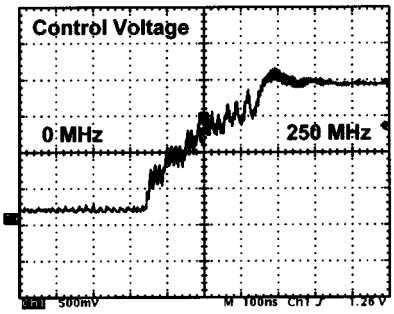


Fig. 15. Control voltage change for a 0–250 MHz frequency input.

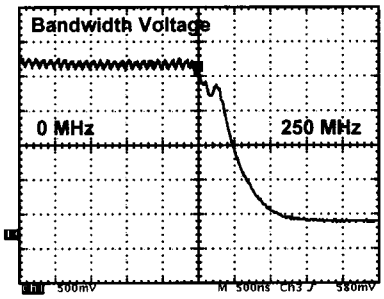


Fig. 16. Loop bandwidth voltage change for a 0–250-MHz frequency input.

In Fig. 13(b), the phase change of the conventional PLL optimized for initial locking is also shown as a dashed line. This simulation result gives several characteristics of the AAPLL. The AAPLL controlled by the recursive algorithm achieves fast lock in the initial locking period, comparable to the speed obtained from a wide-bandwidth PLL because the consecutive error signals rapidly increase the loop bandwidth of the AAPLL. In the steady-state tracking mode, the AAPLL substantially rejects the input jitter due to the narrower loop bandwidth.

V. Experimental Results

The AAPLL is fabricated in a 0.6-μm single-poly triple-metal n-well CMOS process [12]. The die size for the AAPLL is 0.11 mm^2. The total power consumption is less than 15 mW with a single 3-V supply. A microphotograph of the AAPLL is shown in Fig. 14. To get the forgetting factor $\lambda = 0.97$, $R_{\rm BW} = 6.6\,{\rm k}\Omega$, $C_{\rm BW} = 20$ pF are used. And $I_\alpha = 100\ \mu$A is selected to get $\beta = 3 \cdot 10^{-3}$. The locking-speed measurement is carried out using an abrupt change of the input signal frequency from 0 to 250 MHz. Fig. 15 shows the corresponding

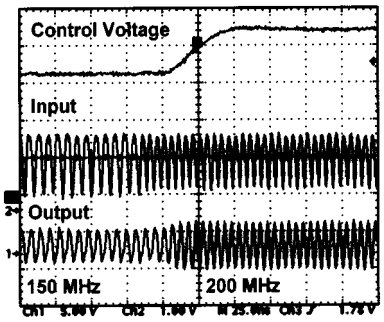


Fig. 17. Experimental results of the locking for a 150–200-MHz input signal.

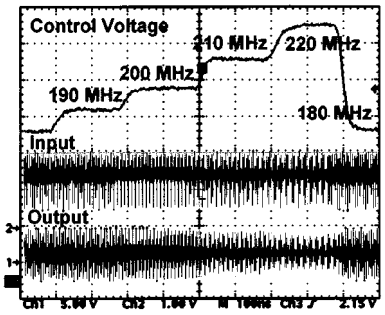


Fig. 18. Experimental results of the locking for a 180–220-MHz input signal by four steps.

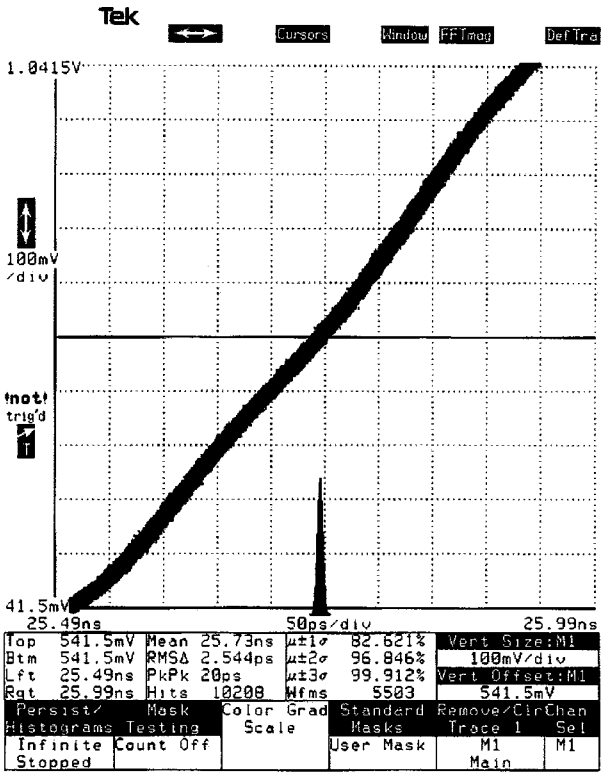


Fig. 19. 2.544-ps (rms) and 20-ps (peak-to-peak) cycle-to-cycle jitter at 250-MHz input.

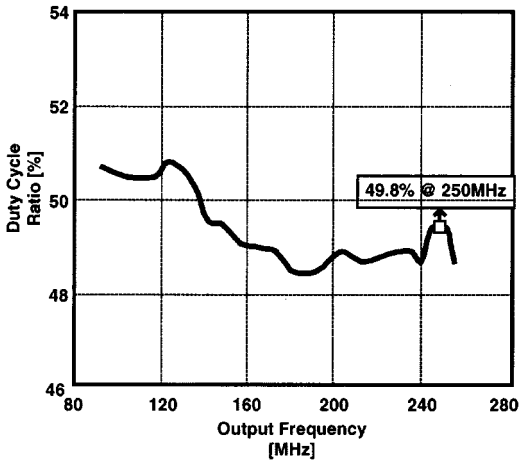


Fig. 20. 50% duty-cycle correction operation over the entire frequency range.

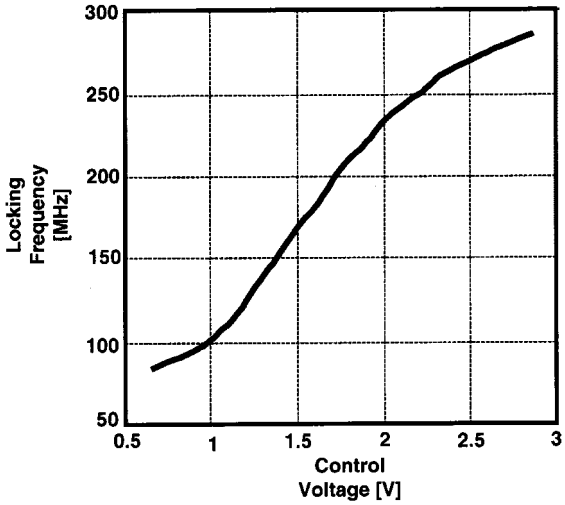


Fig. 21. VCO linearity.

VCO control-voltage variations. In order to measure the locking speed precisely, the running cycles of the output waveform are counted from the frequency triggering point in the initial locking state. The AAPLL requires less than 30 clock cycles for both frequency and phase lock in this case. The voltage of $C_{\rm BW}$, representing the AAPLL bandwidth, is also measured. Fig. 16 shows the measured voltage and describes the adaptation of the loop bandwidth in the AAPLL. Figs. 17 and 18 show the measured control voltages and the corresponding output signals when the input frequencies vary from 150 to 200 MHz and from 180 to 220 MHz by four steps respectively. In this case, the frequency and the phase locking require less than 10 symbol periods because the frequency steps are much smaller compared to the previous case. The measured cycle-to-cycle jitters of the AAPLL output signal at a 250-MHz input signal are 2.54 ps (rms) and 20 ps (peak-to-peak) as depicted in Fig. 19. This jitter value contains the inherent measurement setup jitter [13].

In order to test the performance of the duty cycle correction circuit, the duty cycle ratio of the output signal is measured with the input signals from 90 to 260 MHz. Fig. 20 shows the measured result of the corresponding duty cycle ratio. This result indicates the feed forward-type duty-cycle corrector maintains

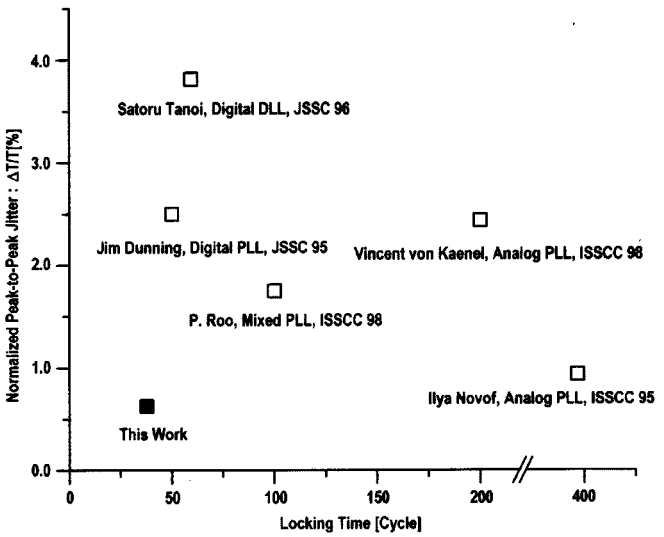


Fig. 22. Comparison between recently reported PLL's and DLL's and this work.

TABLE I
AAPLL CHARACTERISTICS SUMMARY

Technology	0.6-μm *single-poly* 3-level metal n-well Standard CMOS process
AAPLL core size	0.11 mm^2
Measured Cycle-to-Cycle jitter	2.544 ps (RMS) 20 ps (peak-to-peak) at 250-MHz input signal
Power dissipation	15 mW at 3 V 2.3 mW at 1.5 V
Duty cycle ratio	49.8 % at 250-MHz input signal
Locking time	< 30 cycles to lock onto unknown frequency signal < 10 cycles to lock onto frequency-hopping signal

50% duty-cycle ratio within 2% error for the region. Fig. 21 shows a VCO linearity diagram. The VCO gain is about 100 MHz/V at a 250-MHz input frequency. The AAPLL operates from 80 to 290 MHz with a 3-V supply voltage. Fig. 22 compares the normalized peak-to-peak jitter and the lock time of the AAPLL with those of recently reported PLL's and DLL's. Measured characteristics are summarized in Table I.

VI. CONCLUSION

This paper presents the design of a 250-MHz low-jitter fast-lock analog adaptive bandwidth-controlled PLL on a single chip. The chip is implemented in a 0.6-μm standard CMOS process. Simple recursive control logic is proposed to control the bandwidth effectively. The measured locking time is less than 10 cycles in a 10-MHz frequency step and less than 30 cycles from an unknown frequency signal to the 250–MHz signal respectively. The measured output jitters are 2.6 ps (rms) and 20 ps (peak-to-peak). All the components are designed

using analog technique and hence the required die size and the power consumption are minimal.

APPENDIX

As shown in Fig. 2, the OR gate gives the control signal for the switch S according to the phase error signal in the bandwidth controller. When the phase error of the signal is high, the controller signal from the OR gate feeds current to the bandwidth capacitor $C_{\rm BW}$ and the voltage across the capacitor increases at a constant rate $I_\alpha/C_{\rm BW}$. As a result, the bandwidth voltage $V_{\rm BW}$ increases proportional to the normalized phase error $\Delta\theta(n)$. After the charging process, the controller signal from the OR gate disconnects the path from the current source and connects to the resistor. So the capacitor $C_{\rm BW}$ discharges through the resistor $R_{\rm BW}$. The switching action occurs every clock cycle period.

The voltage $V_{\rm BW}$ of the bandwidth capacitor at time $(n+1)T$ can be written as

$$V_{\rm BW}[(n+1)T] = \left[V_{\rm BW}[nT] + \frac{I_\alpha}{C_{\rm BW}} \cdot \frac{T}{2\pi}|\Delta\theta(n)|\right] \times e^{-\frac{T}{R_{\rm BW}C_{\rm BW}}\left(1-\frac{|\Delta\theta(n)|}{2\pi}\right)} \quad (7)$$

where $V_{\rm BW}[nT]$ is the voltage of the previous capacitor voltage at time nT. The voltage equation can be simplified to (8).

$$V_{\rm BW}[(n+1)T] = \lambda(n) \cdot V_{\rm BW}[nT] + \beta(n) \cdot |\Delta\theta(n)| \quad (8)$$

Here $\lambda(n) = \exp[-T/R_{\rm BW}C_{\rm BW}(1-|\Delta\theta(n)|/2\pi]$, $\beta(n) = (I_\alpha/C_{\rm BW}) \cdot (T/2\pi) \cdot \lambda(n)$. In the initial locking mode, the AAPLL does the locking operation based on (8). Once the AAPLL finished the phase and frequency locking, the phase error is far less than 2π. In this case, the forgetting factor and the proportional coefficient can be be replaced by $\lambda = \exp[-T/R_{\rm BW}C_{\rm BW}]$ and $\beta = (I_\alpha/C_{\rm BW}) \cdot (T/2\pi) \cdot \lambda$.

REFERENCES

[1] J. Dunning *et al.*, "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.

[2] B. Kim, D. N. Helman, and P. R. Gray, "A 30-MHz hybrid analog/digital clock recovery circuit in 2-μm CMOS," *IEEE J. Sold-State Circuits*, vol. 25, pp. 1385–1394, Dec. 1990.

[3] M. Mizuno *et al.*, "A 0.18 μm CMOS hot-standby phase-locked loop using a noise immune adaptive-gain voltage-controlled oscillator," *ISSCC Dig. Tech. Papers*, pp. 268–269, Feb. 1995.

[4] G. Roh, Y. Lee, and B. Kim, "An optimum phase-acquisition technique for charge-pump phase-locked loops," *IEEE Trans. Circuit Syst. II*, vol. 44, pp. 729–740, Sept. 1997.

[5] B. Chun, Y. Lee, and B. Kim, "Design of variable loop gain of dual-loop DPLL," *IEEE Trans. Commun.*, vol. 45, pp. 1520–1522, Dec. 1997.

[6] P. F. Driessen, "DPLL bit synchronizer with rapid acquisition using adaptive Kalman filtering techniques," *IEEE Trans. Commun.*, vol. 452, pp. 2673–2675, Sept. 1994.

[7] B. Kim, "Dual-loop DPLL gear-shifting algorithm for fast synchronization," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 577–586, July 1997.

[8] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[9] T. C. Weigandt, B. Kim, and P. R. Gray, "Analysis of timing jitter in CMOS ring oscillators," in *Proc. Int. Symp. Circuit and Systems*, vol. 4, London, U.K., June 1994, pp. 27–30.

[10] C. H. Park and B. Kim, "A low-noise 900-MHz VCO in 0.6-μm CMOS," *IEEE J. Solid-State Circuits*, vol. 34, pp. 586–591, May 1999.

[11] K. Lim, C. H. Park, and B. Kim, "Low noise clock synthesizer design using optimal bandwidth," in *Proc. Int. Symp. Circuit and Systems*, Monterey, CA, June 1998, pp. 163–166.

[12] J. Lee and B. Kim, "A 250 MHz low jitter adaptive bandwidth PLL," *ISSCC Dig. Tech. Papers*, pp. 346–347, Feb. 1999.

[13] J. McNeil, "Jitter in ring oscillators," *IEEE J. Solid-State Circuits*, vol. 32, pp. 870–879, June 1997.

Joonsuk Lee (S'99) received the B.S. and M.S. degrees in electrical engineering and computer sciences from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1995 and 1997, respectively. Since 1997 he has been working toward the Ph.D. degree at the same university.

From 1999 to 2000, he was with IBM Microelectronics, Boston, MA, as an Analog and Mixed Signal Designer involved in a high performance sigma–delta ADC/DAC project with Motorola, Lowell, MA. His research interests include PLL/DLL, timing recovery algorithms, high-speed SDRAM interface, and LAN and mixed-mode signal processing technique for telecommunication IC's.

Mr. Lee is the Gold Medal winner of the Human-Tech Thesis Prize from Samsung Electronics Co. Ltd. in 1997, the Gold Medal winner of the Chip Design Contest from LG Semicon Co. Ltd. in 1998, and the Gold Medal winner of the Integrated Design Center (IDEC) Award in 1998.

Beomsup Kim (S'87–M'90–SM'95) received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1990.

From 1986 to 1990, he worked as a Graduate Researcher and Graduate Instructor at Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. From 1990 to 1991, he was with Chips and Technologies, Inc., San Jose, CA, where he was involved in designing high speed-signal processing IC's for disk drive read/write channels. From 1991 to 1993, he was with Philips Research, Palo Alto, CA, where he was conducting research on digital signal processing for video, wireless communication, and disk drive applications. During 1994, he was a Consultant, developing the partial-response maximum likelihood detection scheme of the disk drive read/write channel. In 1994, he became an Assistant Professor with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, and is currently an Associate Professor. During 1999, he took a sabbatical leave and stayed at Stanford University, Stanford, CA, and also consulted for Marvell Semiconductor Inc., San Jose, CA, on the Gigabit Ethernet and wireless LAN DSP architecture. His research interests include mixed-mode signal processing IC design for telecommunications, disk drive, local area network, high-speed analog IC design, and VLSI system design.

Dr. Kim is a corecipient of the Best Paper Award (1990–1991) for the IEEE JOURNAL OF SOLID-STATE CIRCUITS, and received the Philips Employee Reward in 1992. Between June 1993 and June 1995, he served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING.