Design, Modeling, and Test of a Programmable Adaptive Phase-Shifting PLL for Enhancing Clock Data Compensation

Dong Jiao, Member, IEEE, Bongjin Kim, Member, IEEE, and Chris H. Kim, Senior Member, IEEE

Abstract—Timing compensation between the clock period and datapath delay in the presence of resonant supply noise has drawn a great deal of attention from the circuit design community. This effect, which is often referred to as the clock data compensation effect, manifests itself as an increase in maximum operating frequency for high performance microprocessors. In this work, we propose an adaptive phase-shifting PLL that can achieve optimal clock data compensation by digitally programming the supply noise sensitivity and the phase shift of the PLL clock period. Measurement results from a 1.2 V, 65 nm test chip demonstrate a 3.4-7.3% improvement in the maximum operating frequency across different clock distribution designs and resonant frequencies. A mathematical framework for simulating the performance of the adaptive phase-shifting PLL is presented for better insight on how the proposed PLL performs when used in different clock network configurations. In addition, the impact of the proposed technique on PLL stability as well as its effectiveness in a 32 nm process has been explored.

Index Terms—Resonant noise, adaptive PLL, adaptive clock, clock data compensation.

I. INTRODUCTION

OWER supply noise is considered to be one of the major performance limiting factors in modern low voltage processors [1]. A myriad of solutions to minimize the impact of supply noise on processor performance have been deployed including on-chip decoupling capacitors, resonant damping resistors [2], [3], supply grid optimization techniques, and noise tolerant clock network designs [4]–[9]. Recently, supply noise in the resonant frequency band has been identified as the dominant noise component in high performance designs [10], [11]. Resonant supply noise is caused by the LC tank formed between the package/bonding inductance and the die capacitance and typically resides in the 40 MHz to 300 MHz frequency range [12]. Fig. 1 shows the supply network impedance profiles of IBM's PowerPC[™] chip (left) and Intel NehalemTM microprocessor (right). An impedance peak in the resonant band is clearly shown in both designs [11], [13]. Resonant noise can be excited by either a sudden current spike caused by a

B. Kim and C. H. Kim are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA.

D. Jiao is with Samsung Semiconductor Inc., San Jose, CA 95134 USA (e-mail: dong@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2012.2211171

clock edge or a wakeup/shutdown operation [13], [14]. Once triggered, this so-called "first droop noise" will affect the entire chip manifesting itself as a global supply noise. Due to its large magnitude and relatively long duration, resonant noise constitutes the worst-case supply noise scenario which has triggered a flurry of research activities in the circuit design community [2], [10], [11].

Recent studies have revealed an intriguing timing compensation effect between the clock cycle and the datapath delay in the presence of resonant supply noise [13]–[15]. This phenomenon, often referred to as the "clock data compensation" effect, is illustrated in Fig. 2 in the context of a simple pipeline circuit consisting of a Phase Locked Loop (PLL), a clock path and a datapath. Fig. 2(b) shows an example supply voltage waveform when the resonant noise is excited. The datapath delay depends on the instantaneous supply voltage and as a result, the worst-case datapath delay occurs at point "A". Conventional wisdom says that the clock period must be longer than the sum of the worst-case datapath delay and setup time requirement for correct operation. Fig. 2(b) illustrates this scenario for a supply voltage undershoot event (denoted as "Constant clock period"). In reality however, the PLL output and the clock path delay also gets modulated by the supply noise and may stretch or compress the clock period depending on whether the supply is undergoing a upswing or a downswing. The net effect is a timing compensation between the clock period and datapath delay which helps improve the circuit timing margin. This is also shown in Fig. 2(b) (denoted as "Adaptive clock period") where the varying clock period compensates for the datapath delay variation under resonant supply noise. In other words, a pipeline circuit with an adaptive clock period can operate at a higher frequency than that with a constant clock period.

Adaptive clocking schemes utilizing this principle have been recently proposed to maximize its benefits. One such scheme is to shift the phase of the supply noise seen by the clock path [14], [15], for example by using an RC filtered supply voltage for the clock buffers. This approach has been used in Intel Pentium[™] processors where the supply noise of the clock buffer is reduced by using local RC filters [16]. An alternative way to enhance the clock data compensation effect is by introducing a supply noise sensitive PLL, which has been employed in Intel Nehalem[™] processors [13]. Here, a PLL-based clock generator is designed where the clock period is intentionally tracks the resonant noise.

Previous papers have clearly demonstrated that the clock-data compensation effect is large enough that it can be utilized effectively for improving microprocessor operating frequency. This work attempts to further improve the effectiveness by adaptively programming key parameters such as the phase difference

Manuscript received March 12, 2012; revised July 11, 2012; accepted July 13, 2012. Date of publication September 04, 2012; date of current version October 03, 2012. This paper was approved by Associate Editor Stefan Rusu.



Fig. 1. Supply network impedance of IBM PowerPC[™] (left) and Intel Nehalem[™] processors (right).



Fig. 2. (a) Simplified diagram of a pipeline circuits. (b) Illustration of the clock data compensation effect.

between the supply noises seen by the clock path and the datapath as well as the clock period's sensitivity to supply noise. By doing so, we can achieve the optimal clock data compensation under varying PVT conditions across different clock network topologies. Results from a 65 nm test chip show a 3.4–7.3% improvement in maximum operating frequency for a typical pipeline circuit for supply noise frequencies between 40 MHz and 300 MHz.

II. OVERVIEW OF CLOCK DATA COMPENSATION EFFECT

In this section, we provide an overview of the clock data compensation effect, introduce previous techniques for enhancing this effect, and describe the requirements of phase shift and supply sensitivity for optimal compensation.

A. Definition of Timing Slack

We first define the term "timing slack" in the context of a standard flip-flop based pipeline shown in Fig. 3. To guarantee correct operation, a certain amount of setup time margin must be ensured so that the final outputs arrive at the next flip-flop stage before the next clock edge. Therefore, "slack" is defined as the



Fig. 3. Definition of timing slack in a standard pipeline circuit.

clock period $T_{\rm CLK}$ minus the actual datapath delay $T_{\rm DATA}$. That is

$$slack = T_{\rm CLK} - T_{\rm DATA}.$$
 (1)

B. Existing Techniques for Enhancing Clock Data Compensation Effect

A numerical model was proposed in [14], [15] to quantitatively describe the timing compensation between clock and data. As shown from the modeling and simulation results in [15], there exists an intrinsic "beneficial" compensation effect in typical pipeline circuits. In other words, the clock period variation usually helps improve the timing slack. Simulation results in [15] also indicate that the clock data compensation can be enhanced by optimizing the clock path delay or its sensitivity to supply noise.

In real designs however, the clock path delay is not something that can be set arbitrarily due to other design constraints such as skew, slew, and power consumption. Non-intrusive methods have been preferred such as adaptive clocking schemes in which the clock period is modulated by the supply noise so that the compensation effect can be enhanced. For example, Intel Pentium[™] processors utilized this timing compensation effect by applying a separate RC low-pass filtered supply voltage for the clock buffers [16]. The low-pass filter determines the phase and the amplitude of the supply noise seen by the clock buffers, maximizing the clock data compensation effect. In [15], a stacked buffer with built-in RC filters was proposed enabling similar



Fig. 4. Illustration of adaptive clocking schemes for clock data timing compensation.

control of the phase and the amplitude of the supply noise while reducing the area overhead incurred by the large capacitors in [16]. Finally, a novel adaptive supply—tracking PLL has been introduced in Intel Nehalem[™] processors [13], in which the output clock period tracks the supply noise to optimize the clock data compensation.

C. Achieving Optimal Clock Data Compensation

As shown in the previous section, several adaptive clocking schemes have been proposed to enhance the timing compensation between clock cycle and datapath delay. One natural question here is whether the existing approaches can achieve optimal compensation across different clocking network designs under severe PVT variations. To answer this question, let us first perform a brief analysis of the adaptive clocking scheme as shown in Fig. 4. The four waveforms represent (i) the resonant supply noise and the clock period modulation effect seen by (ii) the PLL, (iii) the clock network and (iv) the local flip-flops, respectively. The minimum supply voltage occurs at "A", which is also the point when the datapath delay is the worst. Suppose the adaptive PLL produces the longest clock period at "B" [13] and the clock cycle is stretched to its maximum at "C" when the supply voltage has the sharpest negative slope. Since the clock cycle is modulated by both the PLL and the clock path, the net effect results in the maximum clock cycle occurring somewhere between "B" and "C", denoted as "D". Once we account for the clock path delay, the local flip-flops see the maximum clock cycle at "E". To achieve optimal timing compensation between the clock cycle and the datapath delay, "E" needs to be aligned with the maximum datapath delay ("A") with the same phase and amplitude. Therefore, an additional phase shift and proper adjustment of the clock period's sensitivity to supply noise are required for the best possible timing compensation, as shown as "Bopt". Previous designs, however, did not consider these effects and were not able to adapt to different design parameters. Inspired by these observations, we propose an adaptive phase-shifting PLL design, in which both the phase shift and the supply noise sensitivity of the clock can be digitally programmed for the optimal performance.

III. MODELING OF CLOCK DATA COMPENSATION EFFECT

Both analytical models and numerical methods have been proposed to estimate the clock data compensation effect [14], [15]. In this section, we will briefly describe how to apply the numerical method for analyzing the clock data compensation effect in various non-adaptive and adaptive clocking schemes. Readers are encouraged to read [14], [15] for further details on the modeling methodology.

A. Derivation of Timing Model

To model the clock data compensation effect, a digital signal in a clock path or a datapath is treated as a travelling wave propagating through a fixed length medium. The velocity of this wave is proportional to the instantaneous supply voltage and can be expressed as

$$v(t) = SA_0 + sa\cos(\omega_m t - \theta) \tag{2}$$

where A_0 and a are the DC and AC amplitudes of the supply voltage, S and s are the AC and DC sensitivities of v(t) with respect to the supply voltage, ω_m is the noise frequency and θ is the noise phase [14]. Let Y_0 be the total physical distance travelled by a clock edge as it propagates through the clock network. Then, we can express Y_0 as the integration of the clock edge's velocity over the total travelling time t_e . Also note that Y_0 is proportional to the nominal delay D_0 , which gives

$$Y_0 = D_0 S A_0 = \int_0^{t_e} \left[S A_0 + sa \cos(\omega_m t - \theta) \right] dt.$$
 (2)

As suggested in [15], numerical methods are needed to solve (3) in order to obtain an accurate t_e .

Next we will use a standard flip-flop based pipeline circuit shown in Fig. 3 to describe the steps for deriving the timing slack using this numerical model. Suppose the first clock edge E_1 launched from the clock generation block at time t = 0 takes t_{cp1} to reach the flip-flop. The input data of the first flip-flop starts to propagate through the datapath at time $t = t_{cp1}$ and takes t_d to reach the input of the second flip-flop. Now assume the second clock edge E_2 is launched at time $t = t_{clk}$ and takes t_{cp2} to propagate through the clock path. Then, the timing slack can be calculated as

$$slack = t_{clk} + t_{cp2} - t_{cp1} - t_d.$$
 (4)

Clearly, t_{clk} , t_{cp2} , t_{cp1} and t_d need to be solved in order to calculate the timing slack. Similar to (3), the following set of equations can be used to derive timing parameters t_{clk} , t_{cp2} , t_{cp1} and t_d :

$$T_{clk} = \int_{0}^{t_{clk}} \left[S_{PLL} V_{DD} + s_{PLL} v_{DD} \cos(\omega_m t - \theta_0 - \theta_{PLL}) \right] dt$$

$$T_{cp} = \int_{0}^{t_{cp1}} \left[S_{cp} V_{DD} + s_{cp} v_{DD} \cos(\omega_m t - \theta_0 - \theta_{cp}) \right] dt$$

$$T_{cp} = \int_{t_{cp1}}^{t_{cp1} + t_{cp2}} \left[S_{cp} V_{DD} + s_{cp} v_{DD} \cos(\omega_m t - \theta_0 - \theta_{cp}) \right] dt$$

$$T_d = \int_{t_{cp1}}^{t_{cp1} + t_d} \left[S_d V_{DD} + s_d v_{DD} \cos(\omega_m t - \theta_0) \right] dt$$
(5)

Here, T_{clk} , T_{cp} and T_d are the clock period, the clock path delay and the datapath delay, respectively, under a nominal supply voltage. θ_{PLL}/θ_{cp} represents the phase difference between the



Fig. 5. Dependency of the worst-case slack on phase shift (θ_{PLL}) and supply noise sensitivity (s_{PLL}).

supply noise seen by the datapath and the PLL/clock path. In a conventional pipeline circuit design, θ_{PLL} and θ_{cp} are both 0 while non-zero θ_{PLL} and θ_{cp} values are used in adaptive PLL designs or phase-shifting clock buffer designs. θ_0 is the arbitrary initial phase when the first clock edge arrives at the first flipflop. Since we are interested in the worst-case slack, (4) and (5) need to be solved numerically by sweeping θ_0 from 0 to 2 π and taking the minimum value as the worst-case timing slack. One thing to note here is that these four equations can be used to model both the phase-shifting PLL design as well as the phase-shifted clock distribution design. For example, the effectiveness of the former can be estimated by adjusting s_{PLL} and θ_{PLL} while the effectiveness of the latter can be verified using different s_{cp} and θ_{cp} values.

B. Modeling of Adaptive Clocking Schemes

As discussed in Section II.C, the phase shift (θ_{PLL}) and the supply noise sensitivity (s_{PLL}) of a phase-shifting PLL design need to be carefully chosen in order to achieve the optimal clock data compensation. In this section, we apply the aforementioned numerical model to provide a deeper insight into various adaptive clocking schemes. For this experiment, the clock path delay of the circuit under test is set as 1.0 ns while the clock period and datapath delay under a nominal supply voltage are both 0.83 ns. Fig. 5 shows the dependency of the worst-case timing slack on the phase shift (θ_{PLL}) and the supply noise sensitivity (s_{PLL}) for two different clock distribution designs. In the first test, the frequency of the resonant supply noise is set as 150 MHz while the clock distribution under test includes a large RC filter which reduces the supply noise seen by the clock buffers by 80% [16]. To mimic the impact of the RC filter, s_{cp} and θ_{cp} values are set as 0.2 s_d and 0.435 π (cos⁻¹ 0.2) respectively in the numerical model to account for the impact of phase-shifted clock buffers. As shown in Fig. 5(left), the optimal slack can be achieved when $s_{\rm PLL} = 1.0 \, s_d$ and $\theta_{\rm PLL} = 0.3 \pi$. In the second test, the resonant noise is set to 40 MHz while the clock distribution is assumed to be a chain of normal buffers with long interconnect in between them. In practice, the interconnect limited clock path usually has a lower sensitivity compared with a datapath [14], so in this test, we set the sensitivity of the clock path 30% lower than the datapath sensitivity ($s_{cp} = 0.7 s_d$ and $\theta_{cp} = 0$). Simulation results of the worst-case slack are given in Fig. 5(right) showing an optimal configuration at $s_{PLL} = 1.05 s_d$ and $\theta_{PLL} = 0.05 \pi$. As shown in Fig. 5, the optimal configuration can vary significantly depending on the clock distribution design, resonant frequency, and so on. These results again confirm the need for programmable phase shift and supply noise sensitivity in order to achieve the optimal performance under a wide range of operating conditions.

We also apply the numerical model to several other clock distribution designs with different characteristics (i.e., θ_{cp} and s_{cp}) and the results are summarized in Table I. The optimal $\theta_{\rm PLL}$ and $s_{\rm PLL}$ of the adaptive phase-shifting PLL design depends on the clock distribution characteristics. It is interesting to look into the extreme case when there is no supply noise in the clock distribution (clock tree #4). As expected, the maximum clock period point needs to be shifted by 1.0 ns (=clock path delay) so that it can compensate the maximum datapath delay point. Since the noise frequency is 80 MHz, the desired phase shift can be easily calculated as 0.16π , which is consistent with the modeling result (0.17π) . Another interesting case is for clock trees where the clock and data paths have the same supply noise sensitivity. The modeling results for clock trees #5, #6 and #7 show that no phase shift is needed for different resonant frequencies. This interesting result can be qualitatively explained as follows: the worst-case datapath delay occurs when the supply voltage is minimal. The corresponding clock cycle seen by local flip-flops, on the other hand, is affected by both the PLL and the clock path. In an adaptive PLL, the maximum clock is generated when the supply voltage is the minimum. This maximum clock propagates through the clock path, causing its arrival time at the local flip-flops to fall behind the worst-case datapath delay point. Meanwhile, the clock path modulates the clock signal, leading to the maximum stretch point 0.5π ahead of the worst-case datapath delay point (refer to Fig. 4). These two effects compensate for each other, and as a result, when the PLL and the clock path have the same supply noise sensitivity, the maximum clock coincides with the worst-case data path delay point regardless of the noise frequency. We can also see that by choosing the optimal configuration for the proposed

TABLE I Optimal Configurations and Performance of the Proposed PLL for Different Clock Distribution Designs ($f_{clk} = 1.2 \text{ GHz}$, $T_{cp} = 1 \text{ Ns}$)

Clock	Supply	Clock pat	th property	Optim. P	LL config.	Worst-case	Worst-case
tree	noise	$ heta_{cp}$	s_{cp}/s_d	θ_{PLL}	s _{PLL} /s _d	slack w/	slack w/
design	frequency					conv. PLL	new PLL
#1 [16]	150 MHz	0.44π	0.2	0.30π	1	-190 ps	-5 ps
#2	40 MHz	0	0.7	0.05π	1.05	-204 ps	-5 ps
#3 [17]	200 MHz	0.20π	0.81	0.15π	0.5	-58 ps	-16 ps
#4	80 MHz	0	0	0.17π	1	-203 ps	-4 ps
#5	40 MHz	0	1	0	1	-202 ps	-0.3 ps
#6	120 MHz	0	1	0	1	-176 ps	-0.4 ps
#7	300 MHz	0	1	0	1	-126 ps	-0.6 ps



Fig. 6. Schematic of the proposed adaptive phase-shifting PLL design.

PLL, the worst-case timing slack can be improved by 42-201 ps, which is equivalent to 5-24% of the nominal clock period.

IV. PROGRAMMABLE ADAPTIVE PHASE-SHIFTING PLL DESIGN

Fig. 6 shows the schematic of the proposed phase-shifting PLL consisting of a frequency-phase detector, a charge pump, a low-pass filter, a supply tracking modulator, a differential voltage-controlled oscillator (VCO) and a frequency divider. The phase shift and noise sensitivity adjustment are implemented with the supply tracking modulator that consists of three binary-weighted capacitor banks and a bias generation circuit. As can be seen from the schematic, the capacitor banks and transistors M1 and M2 form a high-pass filter so that the resonant supply noise can be AC coupled to the bias voltage of the VCO (VCN, VCP) to generate an adaptive clock signal. Using a proper configuration of the three capacitor banks, the desired phase shift and noise sensitivity can be achieved.

Further details on the capacitor bank operation are provided in Fig. 7. Here, the relationship between the supply noise DVDD (see Fig. 6) and the VCO control signal VCN is derived



Fig. 7. Analysis of the capacitor banks with using Thevenin's theorem.

by Thevenin's theorem using an equivalent voltage source V_{eq} with an equivalent impedance of Z_{eq} . The values of V_{eq} and Z_{eq} can be obtained by calculating the open-circuit output voltage and the short-circuit equivalent impedance. Fig. 8(b)



65nm, 1.2V, room temp., simulation results

Fig. 8. Simulated VCO control voltage for different capacitor bank configurations.



Fig. 9. PLL stability response with additional pole due to the supply tracking modulator.

and (c) show the circuit schematics used to derive each parameter and the resulting expressions. As shown in Fig. 7(d), the equivalent capacitance and the clock period's sensitivity to supply noise can be expressed as $C_{eq} = C_f ||(C_u + C_d)$ and $S_V = C_u/(C_u + C_d)$, respectively, which are both digitally programmable using the capacitor bank. Simulation results in Fig. 8 show the VCO control voltage VCN (see Fig. 7) versus capacitor ratios confirming that the supply noise sensitivity and the phase shift can be varied effectively. Based on the

equations shown in Fig. 7(d), the capacitor values C_u , C_d and C_f can be changed together for the target phase shift and supply sensitivity. The high-pass filter shown in Fig. 7 can introduce a maximum phase shift of 0.5π . We show two extreme cases to justify why this is sufficient for practical applications. First, as discussed in Section III, if the clock path and the datapath have the same supply noise sensitivities, no extra phase shift is required for the optimal compensation (i.e., $\theta_{opt} = 0$). Second, when the clock path sensitivity is 0, the



-150

-200

-250

-300 1M

Phase (deg)



Fig. 11. Differential and RC filtered buffers used in the clock networks.

optimal compensation can be achieved by shifting the supply noise seen by the PLL by $T_{\rm cp}$ (i.e., the clock path delay). So we have $\theta_{opt} = 2\pi * T_{cp} * f_{res}$. In today's high frequency microprocessors, the clock path delay is typically under 1 ns [14] and the resonant frequency is no higher than 300 MHz, so the equation above gives $\theta_{\rm opt} = 0.6\pi$ as the upper bound. Considering that the supply noise sensitivity of practical clock paths is usually much larger than 0, a maximum of 0.5π phase shift introduced by the high pass filter is sufficient for most practical designs.

V. STABILITY ANALYSIS OF ADAPTIVE PLLS

One of the major design challenges for adaptive PLLs is to ensure stable operation. The supply tracking modulator in Fig. 6

Fig. 12. Frequency response of the local supply noise monitor in Fig. 11.

100M

Frequency (Hz)

1G

10G

10M

includes three capacitor banks for coupling the resonant supply noise (DVDD) into the VCO bias generation circuit. The highpass filter formed between the three capacitor banks and a current bias allows only the resonant supply noise to propagate through while suppressing other noise components. On the other hand, the low-pass filter in a typical PLL feedback loop will filter out high frequency components. Therefore, the size of the capacitor banks must be carefully chosen to guarantee good supply noise tracking capability as well as stable PLL locking behavior. To ensure proper tracking operation, the high-pass



Fig. 13. Measured BER versus clock frequency (left). Example supply noise waveforms generated by noise injection circuits (right).

filter should be designed to have a bandwidth lower than the resonant frequency of the chip (e.g., 40 MHz for our system). For stable locking operation of the PLL, the additional pole introduced by the high-pass filter should be carefully determined so that the phase-margin (PM) is greater than 45 degrees.

Three poles exist in a typical charge pump based PLL with the first two located at 0 Hz [18]. The locations of the remaining pole and zero are determined by various PLL design parameters. Fig. 9(a) shows a simple stability plot of a conventional charge-pump PLL. To ensure stable operation, the bandwidth of the PLL should be higher than the zero in order to keep the phase margin larger than 45 degree. One thing to note here is that the PLL stability will be degraded because of the third pole introduced by the parallel capacitor in the loop filter used for suppressing high-frequency ripples. To optimize the PLL stability, we can place the PLL bandwidth at the center of the zero and the third pole frequencies.

In a phase-shifting PLL, the additional pole introduced by the high-pass filter becomes another source of stability degradation. Fig. 9(b) shows this scenario where the additional pole has a frequency between the zero frequency and the third pole frequency. Since the additional pole is in the vicinity of the third pole, the phase margin will decrease $2 \times$ faster than the conventional case as the frequency is increases. To avoid such severe stability degradation and ensure a phase margin no less than 45 degree, we set the frequency of the additional pole to be 10 times larger than the third pole. For example, the PLL can be designed with a 1 MHz bandwidth while its zero and third poles are located at 250 kHz and 4 MHz, respectively. In this case, even after adding a supply tracking modulator for a 40 MHz resonant noise, there will be little stability degradation as the corner frequency of the high-pass filter is sufficiently higher than that of the third pole.

VI. TEST CHIP DESIGN AND MEASUREMENT RESULTS

A. Test Chip Organization

A 1.2 V, 65 nm test chip was designed to verify the effectiveness of the proposed PLL (Fig. 10). The adaptive clock signal

is generated by the PLL and then propagates through the clock network. We have implemented eight different clock trees using regular inverters, differential buffers or RC-filtered buffers [15] with different interconnect lengths. The schematic of the differential buffers and RC-filtered buffers are given in Fig. 11, where the RC constant is chosen as 0.6 ns [15]. A separate 40 pF decoupling capacitor (decap) can be enabled to reduce the supply noise seen by the clock trees. The datapath under test consists of two D-flip-flops and both logic-dominated and interconnect-dominated circuit paths. There is also a reference datapath consisting of a short inverter chain in between two D-flip-flops so that the setup time requirement is always satisfied. An XOR gate is used to compare the sampled results from the datapath with the reference data, and any sampling error will generate a pulse at the XOR output, which increments a 10-bit ripple counter. As a result, the transition in the *i*th bit of the counter output (i.e., BER < 9 : 0 >) indicates that 2isampling errors have occurred. By measuring the average period of the counter output and the clock frequency, the bit-error rate (BER) can be conveniently calculated. The noise injection block has individual devices clocked by an on-chip VCO and a clock pattern synthesis circuit. The clock pattern can be selected from 1, 2, 8 or 32 pulses for every 32 clock cycles to emulate a first-droop or a sinusoidal noise waveform. The amplitude of the injected current can also be digitally adjusted by turning on/off parts of the noise injection devices. The test chip also includes an array of linear feedback shift registers for injecting random supply noise. To monitor the on-chip supply noise, an amplifier-based noise sensor is introduced where the AC components of the power supply and ground are taken as the differential inputs. Fig. 12 shows the frequency response of the on-chip supply noise sensor, from which we can see that the sensor provides a nearly flat gain of -2.5 dB in a large frequency range between 3 MHz and 1 GHz. The static power consumption of this sensor is 2.1 mW.

B. Test Chip Measurement Results

Fig. 13(left) shows an example of the BER data measured at different clock frequencies. Without loss of generality, we de-



Fig. 14. Measured results at 1.2 V and 1.0 V showing the F_{max} (@BER = 10⁻⁶) dependency on phase shift and supply noise sensitivity.

fine the maximum operating frequency as the point when the BER is 10^{-6} , and denote it as $F_{\rm max}$ in this paper. Another interesting conclusion we can draw from this figure is that the impact of the noise from those inserted active devices is much smaller than that of the resonant noise in the proposed PLL. In the BER graph, the slope of the curve and the horizontal location of the curve represent the impacts of the device noise and resonant noise, respectively. For example, the BER curve will be steeper with a larger device noise while it will shift towards the left as the resonant noise gets larger. As we can see from Fig. 13, the BER curve shows a considerable horizontal shift depending on the noise configuration while the slope remains relatively constant. This confirms that the resonant noise has a much larger impact than the device noise.

Fig. 14 shows the measured F_{max} while sweeping the phase shift and supply noise sensitivity values. The chip was tested for a supply voltage of 1.2 V and 1.0 V using a sinusoidal noise waveform. Experimental data shows that F_{max} can be improved by more than 5% for both cases when an optimal configuration is chosen. We also see a large discrepancy in the optimal configurations between the two cases (i.e., 1.2 V and 1.0 V). This is because the timing compensation is affected by various design parameters such as clock frequency, clock path delay, noise frequency, and so on. The proposed PLL is flexible and can adapt to different operating conditions and clock network designs by configuring the phase shift and supply noise sensitivity.

The proposed PLL was tested under different supply noise frequencies. For this test, an inverter-based clock tree was chosen and the noise pattern was configured to emulate the first-droop noise. Measurement results in Fig. 13(left) show a 4% F_{max} improvement for noise frequencies between 40 MHz and 300 MHz. As the noise frequency increases, the performance improvement becomes smaller. This is because the clock path delay makes it difficult, or even impossible, for the

adaptive clock to compensate for the datapath delay variation if the noise period is too short. The proposed PLL was also tested under a 1.0 V supply voltage and the results also show similar performance improvement as shown in Fig. 15(right).

Different clock trees were also tested and the results are shown in Fig. 16(left). Here, clock tree names post pended with "_C" have a 40 pF decap enabled in the clock tree supply and "short" or "long" refers to the interconnect length between the clock buffers. For a 74 MHz sinusoidal noise, the F_{max} is consistently improved by 3.4% to 7.3% verifying the flexibility of the proposed design. Another group of tests were carried out with the first-droop noise injected at 37 MHz under a 1.0 V supply voltage. Measurement results in Fig. 14(right) show a 3.3% to 6.8% improvement in F_{max} across different clock tree designs enabled by the proposed adaptive phase-shifting PLL.

The die microphotograph and the performance summary of our 65 nm test chip are shown in Fig. 17.

VII. SCALABILITY AND PVT VARIATION ANALYSIS IN 32 NM

To validate the scalability of the proposed techniques, we designed and simulated an adaptive phase-shifting PLL for several different clock network designs in an industrial 0.9 V, 32 nm CMOS based on high-k metal-gate technology. Fig. 18 shows the schematic of the test circuit consisting of a phase-shifting PLL operating at 2.58 GHz, a 16-stage FO4 inverter chain datapath and a 20-stage clock buffer chain with a nominal delay of 1.0 ns. For easier control of the clock path characteristics, the amplitude and the timing offset of the supply noise seen by the clock path were adjusted to emulate the behavior of clock paths with different s_{cp} and θ_{cp} . Simulation results of the worst-case timing slack for 4 different clock paths are provided in Fig. 19. As shown on the top left of this figure, for the clock path with the same noise sensitivity as the datapath (i.e., $s_{cp} = 1.0 \ s_d$



Fig. 15. Measured Fmax at 1.2 V and 1.0 V for different noise frequencies.



Fig. 16. Measured Fmax at 1.2 V and 1.0 V for different clock trees.



Technology	65nm LP CMOS	Supply voltage	1.2V
Total area	350 x 250 µm²	PLL area	120 x 100 µm²
Regulation frequency	40MHz-300MHz	F _{max} improve- ment	3.4%-7.3%

Fig. 17. Die microphotograph and performance summary of 65 nm test chip.

and $\theta_{cp} = 0.0\pi$), the best timing slack is obtained at the maximum filter capacitance C_{eq} meaning no phase shift is needed in the PLL, which is consistent with the modeling results shown in Table I. Similarly, the performance of the proposed PLL was simulated for three other clock paths. The results confirm that by







Fig. 18. Test circuit setup used for validating the performance of the proposed PLL in 32 nm.

optimizing the filtering capacitance (C_{eq}) and the supply noise sensitivity (S_v) , the worst-case timing slack can be improved by 37–57 ps or 9.6%–14.7% of clock period for various clock trees.

The impact of PVT variation is another important consideration for designing a phase-shifted PLL. In the proposed PLL design, proper configurations of the equivalent capacitance and the supply noise sensitivity are needed in order to adaptively control the output frequency in the presence of resonant supply noise. In a practical system, however, it is impossible to guarantee an optimal configuration due to static and dynamic PVT variations. For example, when a CPU is operating under various load conditions, it can lead to different IR drop and on-chip



Fig. 19. Simulated timing slack in 32 nm with different configurations of the PLL for different clock trees.



Fig. 20. Simulated timing slack in 32 nm with capacitance and sensitivity variations.

temperatures. As a result, the difference in IR drop will cause the sensitivity to vary in the datapath and the clock path, while the temperature variation will affect the VCO gain. Moreover, at the production level, it might be too costly to calibrate every phase-shifting PLL to have optimal settings. Therefore, it is important to validate the performance of the proposed PLL after taking PVT variations into account. Using the same circuit and setup in Fig. 19, we simulated the slack improvement under PVT variation (Fig. 20). The four different colors represent (1) under optimal configuration, (2) 50% variation in the equivalent capacitance (C_{eq}), (3) 20% variation in the supply noise sensitivity (S), and (4) 50% variation in C_{eq} plus 20% variation in S. As shown in Fig. 20, even under large variations like case (4), the proposed PLL can still achieves 45%–78% of the maximum possible improvement (or 6.8%–12.1% of the clock period). These simulation results are consistent with the test chip data. From Fig. 14, we can see that even after severe variation, the proposed phase-shifting PLL design still provides a slack improvement in the amount of 4.8%–13.3% of the clock period.

VIII. CONCLUSION

An adaptive phase-shifting PLL is proposed for enhancing the clock data compensation effect by tuning the supply noise sensitivity and the phase shift of the PLL clock output. A mathematical framework for simulating the performance of the proposed PLL was used to quantify its effectiveness on different clock network designs. A 1.2 V, 65 nm test chip confirms a 3.4–7.3% improvement in the maximum operating frequency for various clock tree designs and for a supply noise frequency range of 40 MHz to 300 MHz. Our theoretical analysis along with the experimental results show that the proposed adaptive phase-shifting PLL with programmable supply noise sensitivity and phase shift can always achieve the optimal clock data compensation regardless of the operating conditions or the specific clock network topology.

REFERENCES

- M. Saint-Laurent and M. Swaminathan, "Impact of power-supply noise on timing in high-frequency microprocessors," *IEEE Trans. Adv. Packag.*, vol. 27, no. 1, pp. 135–144, Feb. 2004.
- [2] X. Hu, W. Zhao, P. Du, A. Shayan, and C.-K. Cheng, "An adaptive parallel flow for power distribution network simulation using discrete Fourier transform," in *Proc. IEEE/ACM Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2004, pp. 125–130.
- [3] J. Xu et al., "On-die supply-resonance suppression using band-limited active damping," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2007, pp. 286–603.
- [4] J. Gu, R. Harjani, and C. Kim, "Distributed active decoupling capacitors for on-chip supply noise cancellation in digital VLSI circuits," in *Symp. VLSI Circuits Dig.*, 2006, pp. 216–217.
- [5] M. Mansuri and C. K. Yang, "A low-power adaptive bandwidth PLL and clock buffer with supply-noise compensation," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 1804–1812, Nov. 2003.
- [6] T. Fischer, J. Desai, B. Doyle, S. Naffziger, and B. Patella, "A 90-nm variable frequency clock system for a power-managed Itanium architecture processor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 218–228, Jan. 2006.
- [7] S. Yasuda and S. Fujita, "Compact fault recovering flip-flop with adjusting clock timing triggered by error detection," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2007, pp. 721–724.
- [8] X. Hu, T. Toms, R. Radojcie, M. Nowak, N. Yu, and C.-K. Cheng, "Enabling power distribution network analysis flows for 3D ICs," in *Proc. IEEE Int. 3D Systems Integration Conf.*, Sep. 2010, pp. 1–4.
- [9] V. Gutnik and A. Chandrakasan, "Active GHz clock network using distributed PLLs," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1553–1560, Nov. 2000.
- [10] J. Gu, H. Eom, and C. H. Kim, "On-chip supply noise regulation using a low power digital switched decoupling capacitor circuit," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, pp. 1765–1775, Jun. 2009.
- [11] E. Hailu, D. Boerstler, K. Miki, J. Qi, M. Wang, and M. Riley, "A circuit for reducing large transient current effects on processor power grids," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2006, pp. 2238–2245.
- [12] D. Wendel *et al.*, "The implementation of POWER7TM: A highly parallel and scalable multi-core high-end server processor," in *IEEE Int. Solid State Circuits Conf. Dig. Tech. Papers*, 2010, pp. 102–103.
- [13] N. Kurd, P. Mosalikanti, M. Neidengard, J. Douglas, and R. Kumar, "Next generation Intel® core[™] micro-architecture (Nehalem) clocking," *IEEE J. Solid-State Circuits*, vol. 44, no. 4, pp. 1121–1129, Apr. 2009.
- [14] K. L. Wong, T. Rahal-Arabi, M. Ma, and G. Taylor, "Enhancing microprocessor immunity to power supply noise with clock-data compensation," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 749–758, Apr. 2006.
- [15] D. Jiao, J. Gu, and C. H. Kim, "Circuit design and modeling techniques for enhancing the clock-data compensation effect under resonant supply noise," *J. Solid-State Circuits*, vol. 45, no. 10, pp. 2130–2141, Oct. 2010.
- [16] N. A. Kurd, J. S. Barkarullah, R. O. Dizon, T. D. Fletcher, and P. D. Madland, "A multigigahertz clocking scheme for the Pentium® 4 microprocessor," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1647–1653, Nov. 2001.

- [17] D. Jiao, J. Gu, and C. H. Kim, "Enhancing beneficial jitter using phaseshifted clock distribution," in *Proc. IEEE Int. Symp. Low Power Electronics and Design (ISLPED)*, 2008, pp. 21–26.
- [18] F. M. Gardner, "Charge-pump phase-lock loops," *IEEE Trans. Commun.*, vol. COM-28, no. 11, pp. 1849–1858, Nov. 1980.



Dong Jiao (S'10–M'11) received the B.S. degree from Tsinghua University, China, in 2006, the M.S. and Ph.D. degree from the University of Minnesota, Minneapolis, in 2009 and 2011, respectively.

He worked as an intern at Seagate in summer 2008 and at Samsung Semiconductor Inc. from October 2010 to June 2011. Since August 2011, he has been with Samsung Semiconductor Inc. working on circuit design techniques for reliability and PVT variation in advanced technology nodes. His research interests include on-chip variation, reliability, power integrity

for mixed-signal ICs and SRAM design.



Bongjin Kim (S'03–M'10) received the B.S. and M.S. degrees in electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2004 and 2006, respectively. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis.

He spent four years in System LSI, Samsung Electronics, Giheung, Korea, from 2006 to 2010, where he performed research on the clock generator circuits for high-speed serial interface PHY transceivers.

From May 2012 to August 2012, He worked as an intern in wireless business at Texas Instruments, Dallas, TX, where he designed a low-power bulk-acoustic wave oscillator circuits. His research interests include VLSI and analog CMOS integrated circuits including PLL for high-performance processor, low-power biomedical front-end circuits and biosensor.



Chris H. Kim (M'04–SM'10) received the B.S. and M.S. degrees from Seoul National University, Seoul, Korea, and the Ph.D. degree from Purdue University, West Lafayette, IN.

He spent a year at Intel Corporation where he performed research on variation-tolerant circuits, on-die leakage sensor design and crosstalk noise analysis. He joined the electrical and computer engineering faculty at the University of Minnesota, Minneapolis, in 2004, where he is currently an Associate Professor. His research interests include

digital, mixed-signal, and memory circuit design in silicon and non-silicon (organic TFT and spin) technologies.

Prof. Kim is the recipient of an NSF CAREER Award, a Mcknight Foundation Land-Grant Professorship, a 3M Non-Tenured Faculty Award, DAC/ISSCC Student Design Contest Awards, IBM Faculty Partnership Awards, an IEEE Circuits and Systems Society Outstanding Young Author Award, ISLPED Low Power Design Contest Awards, and an Intel Ph.D. Fellowship. He is an author/coauthor of more than 100 journal and conference papers and has served as the technical program committee chair for the 2010 International Symposium on Low Power Electronics and Design (ISLPED).