Circuit Modeling and Design Techniques for Efficient Power Delivery under Resonant Supply Noise

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

DONG JIAO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

CHRIS H. KIM

July 2011

© DONG JIAO 2011

Acknowledgements

First and foremost, I wish to thank Prof. Chris H. Kim, my advisor. I am indebted to him for guiding me during my Ph.D. study at the University of Minnesota and pointing me towards my future career path.

Second, I would like to thank my Ph.D. committee: Prof. Ramesh Harjani, Prof. Sachin Sapatnekar and Prof. Antonia Zhai. Your valuable comments and suggestions helped me improve this thesis.

Last but not least, I thank all my colleagues in the VLSI Research Group in the University of Minnesota for our close collaborations and productive discussions. They are: Dr. Jie Gu, Dr. Tony Kim, Dr. John Keane, Kichul Chun, Wei Zhang, Pulkit Jain, Xiaofei Wang, Seunghwan Song, Ayan Paul, Bongjin Kim and Ed Pataky.

Dedication

To my family.

Abstract

Power supply noise has become one of the main performance limiting factors in sub-1V technologies. Resonant supply noise caused by the package/bonding inductance and on-die capacitance has been reported as the dominant supply noise component in high performance microprocessors. Recently, adaptive clocking schemes have been proposed to mitigate the impact of resonant noise. Here, the clock period is intentionally modulated by the resonant noise when it is generated in PLL or propagates through the clock distribution. As a result, the increased clock period partially compensates for the increased datapath delay which is also modulated by the same resonant noise and this is called *clock data compensation effect*, or *beneficial jitter effect*.

This thesis presents a comprehensive study of this clock data compensation effect including an analysis of its dependency on various design parameters. A mathematical framework, including both an analytical model and a numerical model, is also proposed to accurately describe this timing compensation effect.

To achieve optimal timing compensation, a certain amount of phase shift and proper adjustment of the clock period's sensitivity to supply noise are required. Here we also propose phase-shifted clock distribution designs and an adaptive phase-shifting PLL design to enhance the beneficial clock data compensation effect. Compared with conventional approaches, the proposed phase-shifted clock distribution designs save 85% of the clock buffer area while achieving a similar amount of improvement in the maximum operating frequency (F_{max}) for typical pipeline circuits. In the proposed adaptive phase-shifting PLL, both the phase shift and the supply noise sensitivity of the

clock can be digitally programmed and adjusted so that the optimal compensation can always be achieved under different conditions.

Two test chips were fabricated in a 65nm CMOS process for concept verification. Measurement results demonstrate that the proposed phase-shifted clock distribution designs can provide an 8-27% performance improvement in F_{max} for typical resonant noise frequencies from 100MHz to 300MHz and the proposed phase-shifting PLL can provide 3-7% improvement in F_{max} under various operating conditions.

Table of Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Resonant supply noise

Power supply noise is considered to be one of the major performance limiting factors in sub-1V technologies [1]. Supply noise caused by on-chip current introduces delay variation in datapaths, as well as jitter in clock paths. As a result, the launched data from one stage in a pipeline can no longer be guaranteed to be captured by the next clock edge within a given timing window (i.e., the clock cycle) leading to a timing failure [2]. Significant efforts have been made to alleviate the impact of supply noise on timing errors. A popular method to reduce the supply noise is to add passive or active decoupling components. For example, Pant proposed to optimize the placement of decoupling capacitors (decaps) by using activity profiles based on architecture simulators [3]. Xu proposed an active damping circuit to reduce the resonant noise in the supply grids [4]. Gu proposed an active decap circuit to reduce the decap area and power [5]. All of these techniques to regulate supply noise have power and area overhead. Meanwhile, several circuit techniques and design methodologies have been developed to reduce the clock jitter. For instance, Mansuri proposed an adaptive delay compensation circuit for clock buffers to reduce their sensitivity to supply noise [6]. Chen developed closed-form formulas for jitter prediction and proposed a clock buffer chain to minimize the jitter [7]. More recently, adaptive or error correction circuits were developed to perform jitter compensation on-the-fly. Examples include the noise-adaptive delay line used in Intel's

Foxton processor and the error correction flip-flop which can be re-triggered upon the detection of error proposed by Yasuda [8][9].

Recently supply noise in the resonant frequency band has been shown to be the dominant noise component in high performance microprocessor designs [13][14]. Resonant supply noise is caused by the LC tank formed between the package/bonding inductance and the die capacitance and typically resides in the 40MHz to 300MHz frequency band but can be made as low as 7MHz with a dedicated metal-insulator-metal capacitor technology [20]. Fig. 1 shows the measured supply network impedance of an Intel Nehalem microprocessor which exhibits a large impedance peak at around 150MHz [21]. Resonant noise can be excited by a sudden current spike caused by a clock edge or a wakeup operation [21][22]. Once triggered, this so-called "first droop noise" will affect the entire chip. Due to its large magnitude, resonant noise constitutes the worst-case supply noise scenario which has triggered a flurry of research activities in the circuit design community [4] [10][11][12][13][14][15][16][17][18][19].

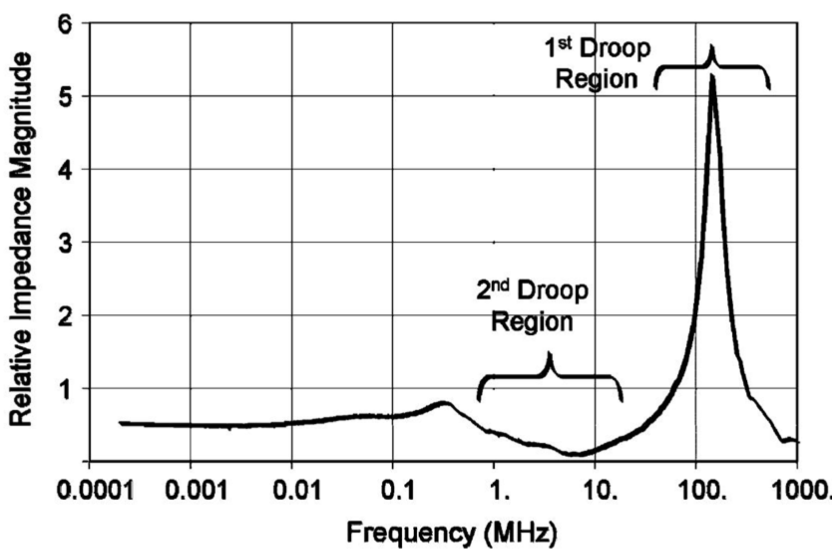


Fig. 1. Measured supply network impedance of Intel’s Nehalem microprocessor [21]

1.2 Clock data compensation effect

Recent papers have revealed an intriguing timing compensation effect between the clock cycle and the datapath delay in the presence of resonant supply noise [21][22][24]. This phenomenon, which is referred to as the clock data compensation effect, or beneficial jitter effect, is illustrated in Fig. 2 with a simple pipeline circuit consisting of a Phase Locked Loop (PLL), a clock path and a datapath. In traditional analysis, the clock period is assumed to be constant and only the datapath delay changes under the influence of supply noise. Fig. 2(b) illustrates example waveforms for this scenario showing several sampling failures during the event of a supply voltage undershoot. In reality, however, the PLL output and the clock path delay may also be modulated by the supply noise and may stretch the clock period during supply downswings. As a result, the varying clock period and datapath delay compensate for each other which could alleviate the timing margin. Fig. 2(c) shows example waveforms for this scenario in which the output is always sampled correctly benefiting from the clock data compensation effect.

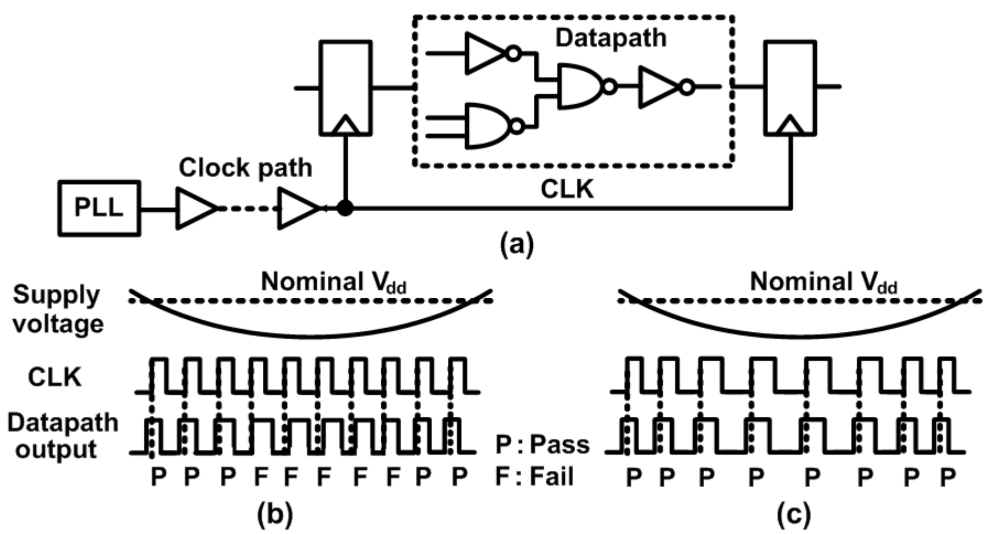


Fig. 2. Illustration of the clock data compensation effect.

Recently, adaptive clocking schemes utilizing this principle have been proposed to enhance the clock data compensation effect. One implementation of this scheme is shifting the phase of the supply noise seen by the clock path [22][24], for example by using an RC filtered supply voltage for the entire clock path. Such an approach has been used in Intel Pentium 4 processors where the supply noise of the clock buffer is reduced by using a local RC filter [25]. 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 [21]. There, a PLL-based clock generator is designed to track the supply noise so that the clock period stretching effect is maximized.

The existing approaches, however, have their own drawbacks and limitations. For example, the local RC filter used in the clock distribution [25] consumes a large silicon area. This is because the resistance in the filter must be small enough to avoid a large IR drop. Therefore, the capacitance has to be large enough to provide a certain amount of phase shift. Moreover, these existing approaches cannot always achieve the optimum clock data compensation because of their limited control on the interactions between the resonant noise and the corresponding adaptive clock. To be more specific, the phase-shifted clock distribution mainly adjusts the phase difference (phase shift) between the supply noises seen by the clock path and the datapath while the supply noise sensitive PLL mainly adjusts the clock's sensitivity to the resonant supply noise. However, as it will be shown later in this paper, both phase shift and supply noise sensitivity need to be carefully adjusted to achieve the optimum compensation under different operating conditions.

In this thesis, we propose phase-shifted clock distribution designs and an adaptive phase-shifting PLL design to enhance the beneficial clock data compensation effect. Compared with conventional approaches, the proposed phase-shifted clock distribution designs save 85% of the clock buffer area while achieving a similar amount of improvement in the maximum operating frequency (F_{max}) for typical pipeline circuits. In the proposed adaptive phase-shifting PLL, both the phase shift and the supply noise sensitivity of the clock can be digitally programmed and adjusted so that the optimal compensation can always be achieved under different conditions. Two test chips were fabricated in a 65nm CMOS process for concept verification. Measurement results demonstrate that the proposed phase-shifted clock distribution designs can provide an 8-27% performance improvement in F_{max} for typical resonant noise frequencies from 100MHz to 300MHz and the proposed phase-shifting PLL can provide 3-7% improvement in F_{max} under various operating conditions.

Chapter 2

CLOCK DATA COMPENSATION EFFECT

In this section, we will first provide the definition of timing slack, and then discuss the impact of clock data compensation effect on both setup time margin and hold time margin. A brief review on the existing techniques for enhancing the clock data compensation effect will be given at the end of this chapter.

2.1 Definition of timing slack

We first define the term timing slack in the context of a standard register-based pipeline shown in Fig. 3. To guarantee correct operations of this circuit, a certain amount of timing margin must be ensured so that the final outputs of the logic block are evaluated before the next clock edge. Therefore, "slack" is defined as the clock period T_{CLK} minus the actual datapath delay T_{DATA}. Obviously, the slack has to be positive for the circuit to be error free. That is:

$$slack = T_{CLK} - T_{DATA} > 0 \tag{1}$$

Here, the setup time requirement is ignored but it can be easily incorporated by adding a timing offset.

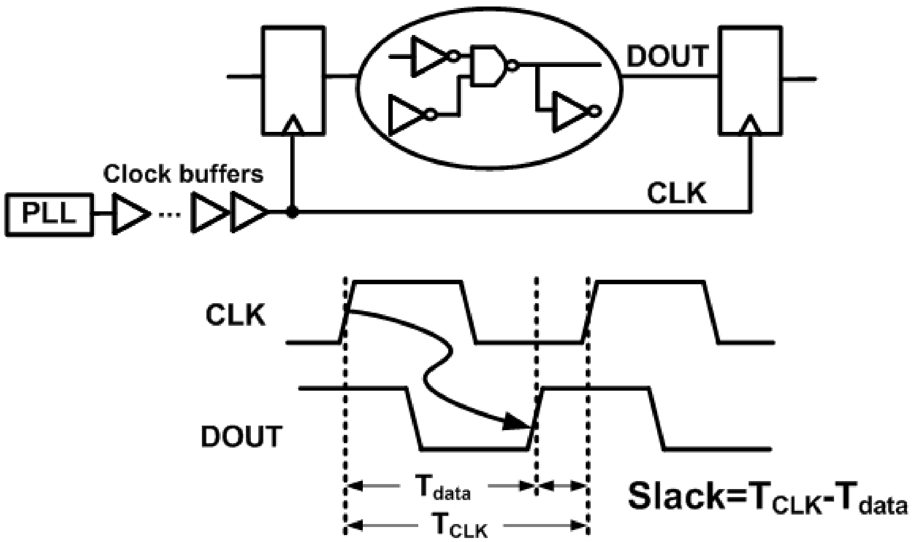


Fig. 3. Definition of timing slack in a standard pipeline circuit.

2.2 Impact of clock data compensation on setup time margin

Conventional analysis only focuses on the increase in datapath delay in the presence of supply noise as shown in Fig. 2(b). However, in reality, the clock path also sees a noisy supply which causes the clock period to gradually stretch during supply downswings (or compression during supply upswings). This clock period modulation effect results in an extra timing margin that compensates for the slowdown in the datapath as shown in Fig. 2(c). Fig. 4 illustrates how the compensation effect improves the setup time margin. In the presence of supply noise, the maximum datapath delay occurs when the supply voltage is at its lowest point, denoted as “A”. The corresponding clock edge (i.e., the 1st edge) which triggers the longest datapath delay signal is launched from the clock source at a certain point in time before “A” as it has to traverse through the clock path. The 2nd edge, which will eventually sample the longest delay signal, is launched one clock period after the 1st edge. It experiences a lower average supply voltage due to the supply downswing, and thus takes a longer time to propagate through the clock path. This makes the clock period longer, compensating for the increased datapath delay.

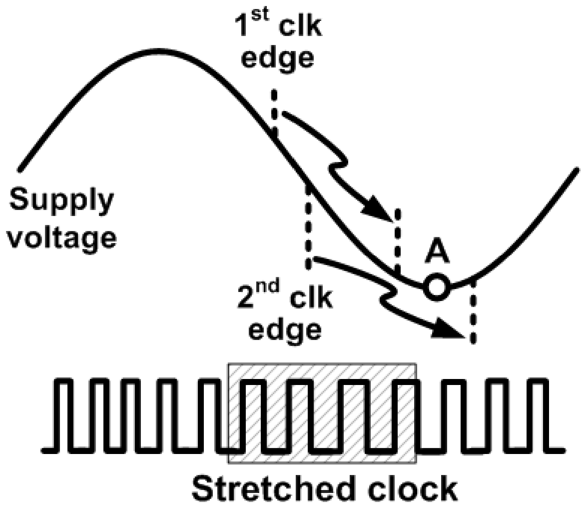


Fig. 4. Setup time margin analysis under resonant supply noise.

2.3 Impact of clock data compensation on hold time margin

Now we discuss how hold time margin is affected by the resonant supply noise. Fig. 5 illustrates the setup and hold time margin requirements for a simple register-based (or latch-based) pipeline. Contrary to the setup time margin scenario, the hold time margin is worst when the datapath delay is minimum, denoted as point “B” in Fig. 6. The corresponding clock edge is triggered when the supply voltage is rising. Here, we only need to consider a single clock edge since hold time violations occur due to clock skew for the same clock edge. As the rising supply voltage compresses the clock period, the clock skew becomes smaller, leading to a minor improvement in the hold time margin as depicted in Fig. 6. This improvement may not be noticeable when considering other timing uncertainties as will be shown in later sections. Note that the analysis on setup time and hold time margins is applicable to both register-based and latch-based designs.

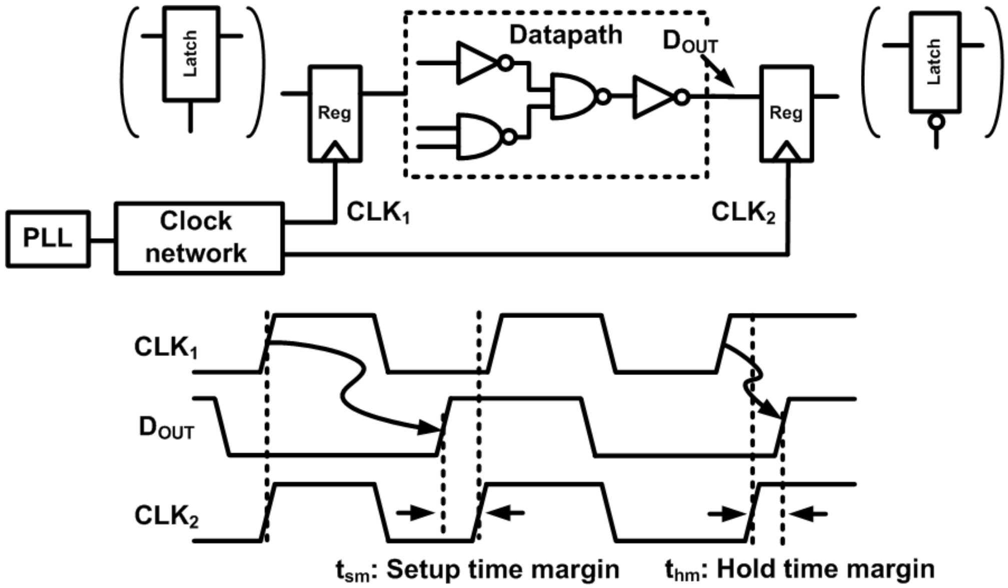


Fig. 5. Illustration of setup and hold time margin in a register-based (or latch-based) pipeline.

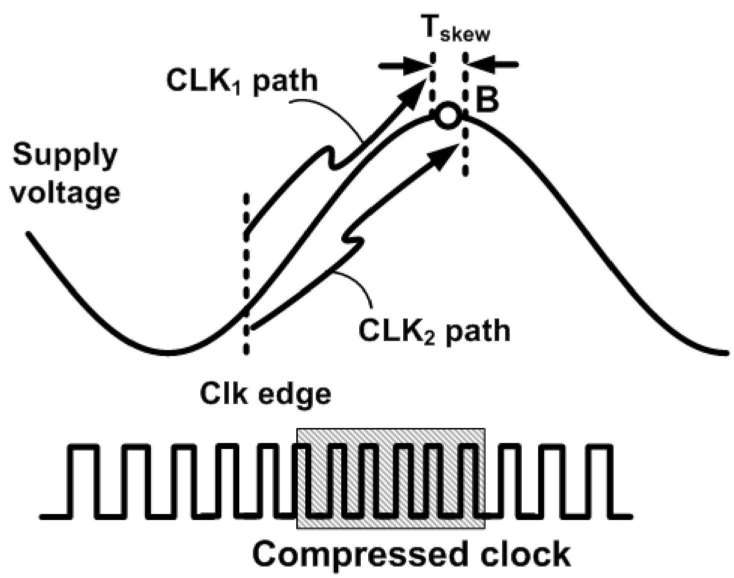


Fig. 6. Hold time margin analysis under resonant supply noise.

2.4 Prior arts for enhancing clock data compensation

Analytical and numerical models have been proposed in [22][24] to quantitatively describe the timing compensation between clock and data. As shown from the modeling and simulation results [24], there exists an intrinsic “beneficial” compensation effect in typical pipeline circuit. In another word, the clock period variation usually helps improve the timing slack. The simulation results from [24] also indicate that the clock data compensation can be enhanced by optimizing the clock path delay or its sensitivity to supply noise.

In reality, however, the clock path delay and its sensitivity to supply noise may not be adjustable since they are usually determined by other design requirements. Therefore, people have proposed adaptive clocking schemes in which the clock period is carefully designed to be sensitive to supply noise so that the compensation between the adaptive clock and the datapath delay can be enhanced. As shown in Fig. 7 (left), [25] proposed using a RC filtered supply voltage for the clock buffers and this technique has been used in Intel Pentium 4 processors. With the help of the low-pass filter, the phase and the amplitude of the supply noise seen by the clock buffers become adjustable so that the clock data compensation effect can be maximized. In [24], a stacked buffer with built-in RC filters has been proposed (Fig. 7 (middle)) enabling similar control on the phase and the amplitude of the supply noise while reducing the area overhead caused by the large capacitors. Fig. 7 (right) shows the schematic of a supply–tracking PLL which has been used in Intel Nehalem processors [21]. In this PLL design, the output clock is designed to be sensitive to the supply noise to optimize the clock data compensation.

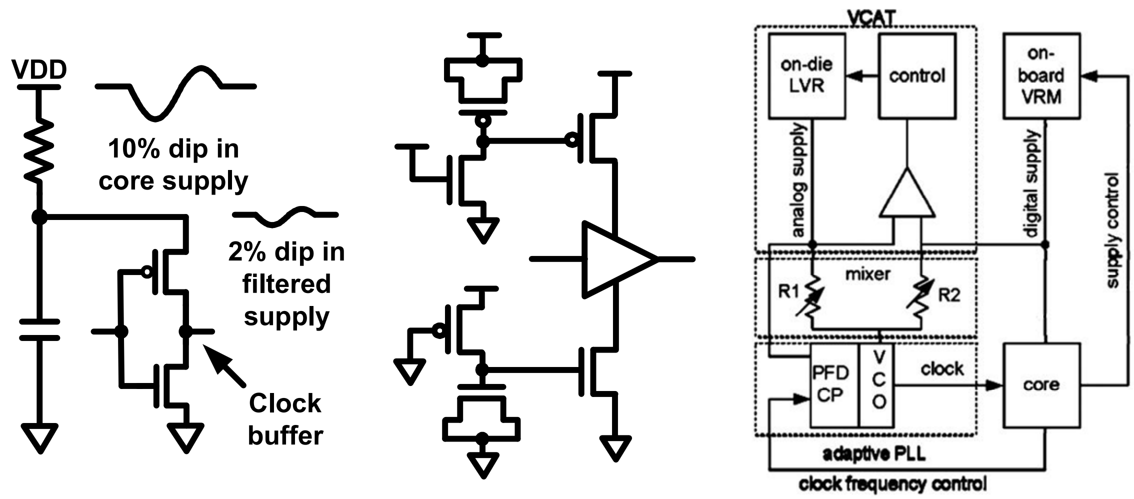


Fig. 7. Phase-shifted clock distribution designs and supply-tracking PLL design.

Chapter 3

MODELING OF CLOCK DATA COMPENSATION

To quantitatively describe the clock data compensation effect, both analytical and numerical models have been proposed [22][24][26]. In this section, details of the derivation and verifications of those models will be provided. We will also explain how to apply those models to various adaptive clocking techniques in order to help circuit designers better understand the timing compensation effect.

3.1 Analytical model

An analytical model for the clock data compensation effect was first derived in [22]. In this section, we will first show the derivation of the analytical mode. As it will be shown later, this model does not match well with HSPICE simulation results due to several simplifications. Therefore, an improved model is derived later which is further verified with simulation results.

3.1.1 Derivation of the analytical model

A signal in a digital circuit (e.g., clock path or datapath signals) can be modeled as a signal wave propagating through a fixed length medium at a velocity which is proportional to the instantaneous supply noise. Fig. 8 illustrates the signal propagation model for the delay on a clock path or a datapath [22].

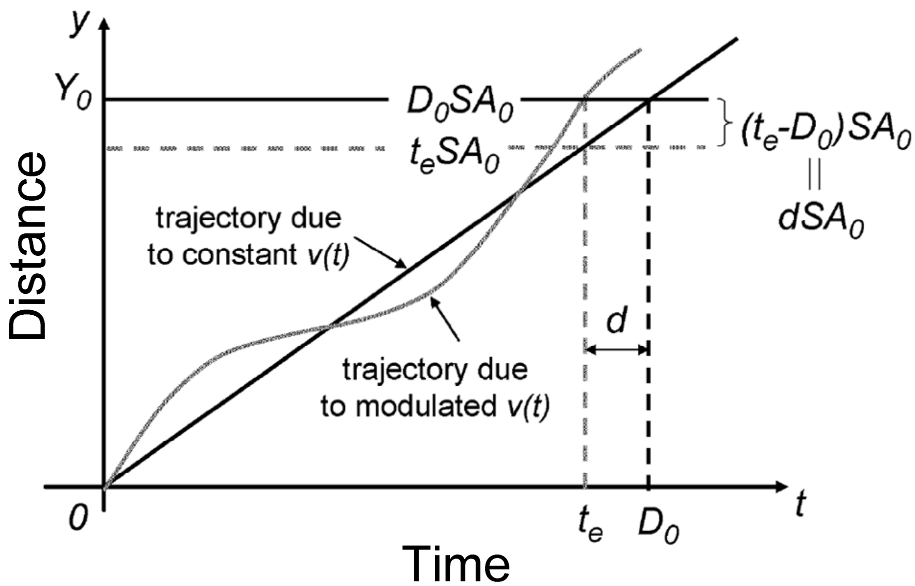


Fig. 8. Delay model for clock path or datapath [22].

The velocity of the traveling wave can be expressed as:

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

where S is the large-signal sensitivity of $v(t)$ with respect to supply, s is the small-signal sensitivity to supply, A_0 is the DC value of supply, a is the AC amplitude of supply, ω_m is the supply noise frequency, and θ is the phase of the supply noise when the signal is issued. Integrating the velocity over the total traveling time t_e gives us the total distance Y_0:

$$Y_0 = D_0 SA_0 = \int_0^{t_e} [SA_0 + sa\cos(\omega_m t - \theta)]dt \tag{3}$$

$$t_e SA_0 + \frac{sa}{\omega_m}(\sin(\omega_m t_e - \theta) - \sin(-\theta)) = D_0 SA_0 \tag{4}$$

Here, D_0 is the nominal traveling time of the signal. By defining the small-signal delay as $d=t_e-D_0$, we get:

$$d = -\frac{2sa}{SA_0\omega_m}\sin\frac{\omega_m t_e}{2}\cos(\frac{\omega_m t_e}{2}-\theta) \tag{5}$$

Using this expression, we can calculate the change in clock period under supply noise by taking the difference between the traveling times of two successive clock edges. The clock period modulation can be calculated as:

$$\Delta p = d[n]-d[n-1] = \frac{4s_{clk}a}{S_{clk}A_0\omega_m}\sin\frac{\theta_{n-1}-\theta_n}{2}\sin\frac{\omega_m t_e}{2}\sin\frac{\omega_m t_e-\theta_n-\theta_{n-1}}{2} \tag{6}$$

where $d[n]$ and $d[n\text{-}1]$ are the traveling time of the n^{th} and $(n\text{-}1)^{th}$ *clock edges* derived from equation (5). θ_n and θ_{n-1} are the phases at which the corresponding clock edges enter the clock path.

Approximating $\theta_n\text{-}\theta_{n-1}=\omega_m/f_{clk}$ and $t_e=D_0=1/f_0$ where $f_{clk}(=1/T_{clk})$ is the clock frequency and f_0 is the inverse of the nominal clock path delay, we find the clock period variation as follows:

$$\Delta p \approx \frac{2s_{clk}af_{clk}}{S_{clk}A_0\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin\frac{\pi f_m}{f_0}\sin(\theta_n-\frac{\pi f_m}{f_0}-\frac{\pi f_m}{f_{clk}}) \tag{7}$$

where Δp has been normalized to the clock frequency f_{clk}.

The datapath delay can be derived similarly using equation (5):

$$d = \frac{-2s_{data}a}{T_{clk}S_{data}A_0\omega_m}\sin\frac{\omega_m t_e}{2}\cos(\frac{\omega_m t_e}{2}-\theta) \approx -\frac{s_{data}a}{S_{data}A}\cos\theta. \tag{8}$$

As it has been derived in [22], here $\omega_m t_e/2$ in the cos() function is ignored because it is relatively small. Finally, the small-signal slack due to clock data compensation can be calculated by finding the difference in the delay variations on the clock path and datapath as follows:

$$slack(\theta) = \Delta p - d = \frac{s_{clk}}{S_{clk}}\frac{2a}{A_0}\frac{f_{clk}}{\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin\frac{\pi f_m}{f_0}\times\sin(\theta-\frac{\pi f_m}{f_0}-\frac{\pi f_m}{f_{clk}})+\frac{s_{data}}{S_{data}}\frac{a}{A_0}\cos\theta \tag{9}$$

Equation (9) was used in [22] as a closed-form solution to evaluate the clock data compensation effect. Note that the second term is the slack caused by delay on the datapath only and has the most negative value of $\frac{s_{data}}{S_{data}}\frac{a}{A_0}$. A negative slack means that the timing margin has been reduced compared with the nominal condition. Thus the design goal is to minimize the most negative (or worst-case) slack in (9).

3.1.2 Proposed analytical model

A simplified clock tree was designed to verify the results from equation (9). A clock path with 26 stages of inverters was used to produce a clock delay of 1ns or f_0 of 1GHz. Another 16 stages of inverters were chained to represent a datapath with a frequency of 2GHz which is also the clock frequency f_{clk}. A supply noise at f_m=200MHz is applied to the supply representing the dominant resonant, or first-droop noise. Because the clock buffers drive interconnects in the datapath, the clock path has lower delay sensitivity with respect to supply noise. $s_{clk}/S_{clk}:s_{data}/S_{data}=0.7:1$ was used in this simulation [22]. Fig. 9 shows that the previous model in (9) exhibits a relatively large discrepancy when compared with HSPICE simulations. The improved worst-case slack due to the beneficial jitter from HSPICE simulation is about 25ps (5% of clock period) which is smaller than the 50ps (10% of clock period) predicted by equation (9). Such a discrepancy comes from several simplifications used during the derivation. Our further evaluation indicates that the approximation of ignoring $\omega_m t_e/2$ in equation (8) introduces a significant error.

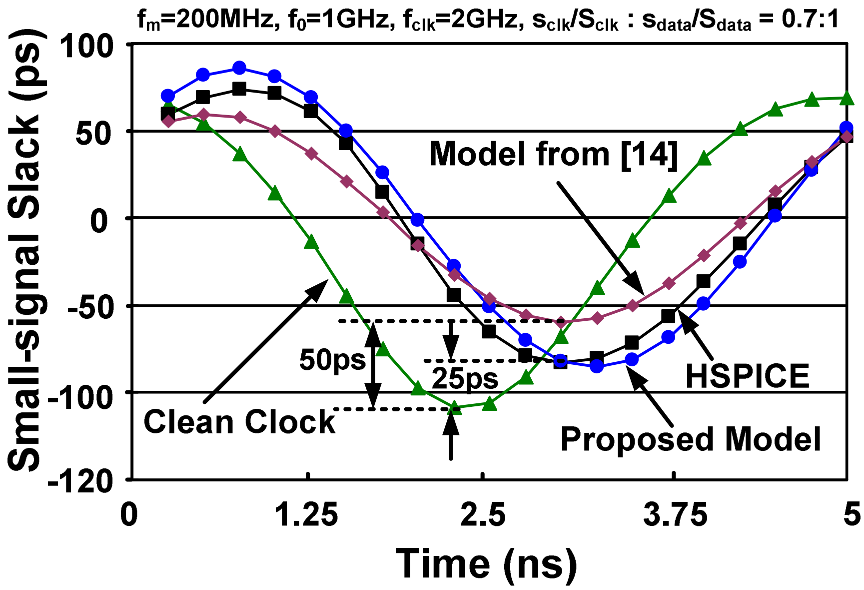


Fig. 9. Slack variation in time domain for different models.

To improve the accuracy of the closed-form model, we consider the term $\omega_m t_e/2$ in (8). As a result, equation (9) becomes:

$$slack(\theta) = \frac{s_{clk}}{S_{clk}} \frac{2a}{A_0} \frac{f_{clk}}{\pi f_m} \sin \frac{\pi f_m}{f_{clk}} \sin \frac{\pi f_m}{f_0} \times \sin(\theta - \frac{\pi f_m}{f_0} - \frac{\pi f_m}{f_{clk}}) + \frac{s_{data}}{S_{data}} \frac{a}{A_0} \cos(\theta - \frac{\pi f_m}{f_{clk}}) \quad (10)$$

Fig. 9 verifies that the slack value predicted from equation (10) has significantly improved the accuracy of the analytical model.

Since θ is a time-varying variable, (10) does not directly indicate the worst-case slack which is most important to a circuit designer. To find the maximum slack values, we convert (10) to:

$$\begin{aligned} slack(\theta) &= \frac{s_{clk}}{S_{clk}} \frac{2a}{A_0} \frac{f_{clk}}{\pi f_m} \sin \frac{\pi f_m}{f_{clk}} \sin \frac{\pi f_m}{f_0} \times (\sin\theta \cos(\frac{\pi f_m}{f_0} + \frac{\pi f_m}{f_{clk}}) - \cos\theta \sin(\frac{\pi f_m}{f_0} + \frac{\pi f_m}{f_{clk}})) \\ &+ \frac{s_{data}}{S_{data}} \frac{a}{A_0} (\cos\theta \cos\frac{\pi f_m}{f_{clk}} + \sin\theta \sin\frac{\pi f_m}{f_{clk}}) \\ &= A\sin\theta - B\cos\theta = \sqrt{A^2 + B^2}\sin(\theta + \phi) \end{aligned} \quad (11)$$

where
$$A = \frac{s_{clk}}{S_{clk}}\frac{2a}{A_0}\frac{f_{clk}}{\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin\frac{\pi f_m}{f_0}\cos(\frac{\pi f_m}{f_0}+\frac{\pi f_m}{f_{clk}})+\frac{s_{data}}{S_{data}}\frac{a}{A_0}\sin\frac{\pi f_m}{f_{clk}}$$
$$B = \frac{s_{clk}}{S_{clk}}\frac{2a}{A_0}\frac{f_{clk}}{\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin\frac{\pi f_m}{f_0}\sin(\frac{\pi f_m}{f_0}+\frac{\pi f_m}{f_{clk}})-\frac{s_{data}}{S_{data}}\frac{a}{A_0}\cos\frac{\pi f_m}{f_{clk}}$$
$$\phi = a\tan(\frac{-B}{A})$$

Now, the worst-case slack in equation (11) can be found from the magnitude of that equation:

$$\left|slack_{wc}\right| = \sqrt{4(\frac{s_{clk}af_{clk}}{S_{clk}A_0\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin\frac{\pi f_m}{f_0})^2+(\frac{s_{data}a}{S_{data}A_0})^2-4\frac{s_{clk}s_{data}}{S_{clk}S_{data}}(\frac{a}{A_0})^2\frac{f_{clk}}{\pi f_m}\sin\frac{\pi f_m}{f_{clk}}\sin^2\frac{\pi f_m}{f_0}} \quad (12)$$

It is important to realize that the interplay between the clock and data can either improve or degrade the timing slack depending on the phase between the signals and the supply noise. If we compare the clean clock and the noisy clock results in Fig. 9, the slack is improved for the earlier noise cycle while for the rest of the time, the slack is actually worsened. However, the compensation between the clock and data is beneficial for the worst-case slack $|slack_{wc}|$ which is more critical. The smaller the $|slack_{wc}|$ is, the less performance degradation the supply noise will inflict. Because f_{clk} (>2GHz) is much higher than f_m (<300MHz), $\sin(\pi f_m/f_{clk})$ can be approximated as $\pi f_m/f_{clk}$. So (12) can be further simplified to:

$$\left|slack_{wc}\right| = \sqrt{4(\frac{s_{clk}}{S_{clk}})(\frac{a}{A_0})^2\sin^2\frac{\pi f_m}{f_0}(\frac{s_{clk}}{S_{clk}}-\frac{s_{data}}{S_{data}})+(\frac{s_{data}}{S_{data}}\frac{a}{A_0})^2} \quad (13)$$

The second term inside the square root of (13) models the slack degradation with a clean clock while the first term models the compensation effect from the clock path. Equation (13) can be used by circuit designers to optimize the effect of the clock data compensation. Because f_m is determined by the package and f_{clk} has always been pushed toward limits, the parameters that can be adjusted to minimize the $|slack_{wc}|$ are clock propagation delay f_0, clock path sensitivity s_{clk}/S_{clk} and datapath sensitivity s_{data}/S_{data}. Equation (13) indicates that compared with a clean clock case, the slack is improved only when $s_{clk}/S_{clk}<s_{data}/S_{data}$, which is usually true because of the interconnect RC in the clock path. Fig. 10 shows the worst-case slack variation versus relative ratio between delay sensitivities of the clock path and the datapath. The result follows the trend predicted by (13). Smaller clock path sensitivity produces better compensation. The minor discrepancy between simulation and model comes from the simplification used when deriving (13).

Furthermore, equation (13) predicts that the maximum compensation happens when:

$$\sin\frac{\pi f_m}{f_0}=1 \quad \text{or} \quad f_0=2f_m \tag{14}$$

This result is consistent with what was shown in [22] and is verified by simulations in Fig. 11. The best clock path delay happens at 400MHz (=2f_m) and improves the worst-case slack by 58ps (12% of clock period) compared with the clean clock case.

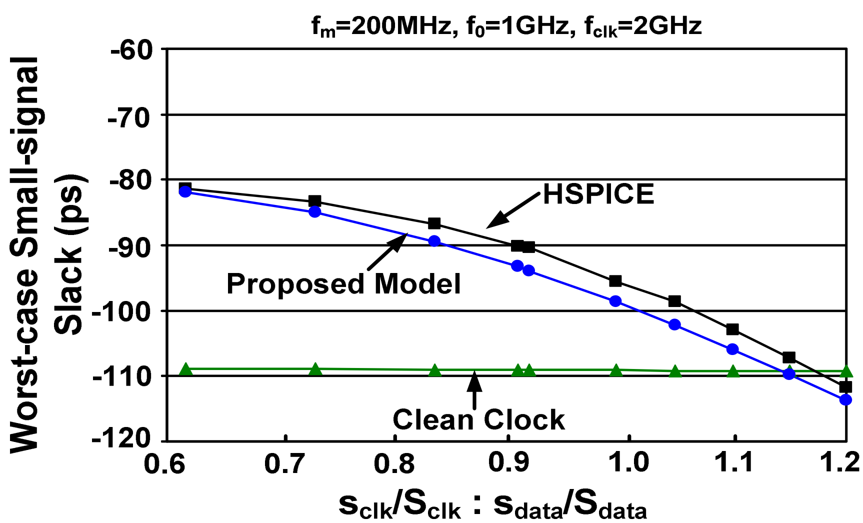


Fig. 10. Worst-case slack variation vs. delay sensitivities.

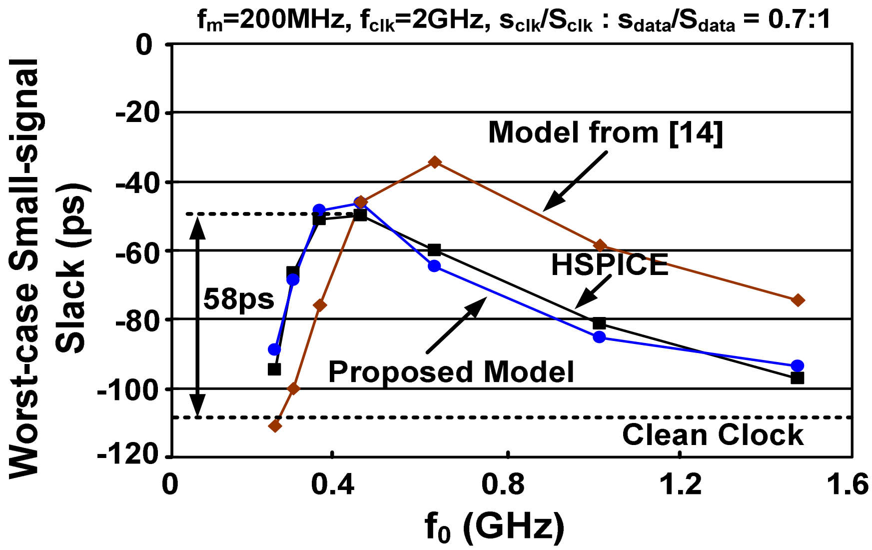


Fig. 11. Worst-case slack variation vs. clock path delay frequency f_0.

3.2 Numerical model

Next we will use a standard register-based pipeline circuit shown in Fig. 3 to describe the flow 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 arrive at the register. The input data of the first register starts to propagate through the datapath at time $t=t_{cp1}$ and takes t_d to reach the input of the second register. 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

$$slack = t_{clk} + t_{cp2} - t_{cp1} - t_d \tag{15}$$

Similar to (3), four equations can be established for t_{clk}, t_{cp2}, t_{cp1} and t_d as follows:

$$
\begin{aligned}
T_{clk} &= \int_0^{t_{clk}} [S_{PLL}V_{DD} + s_{PLL}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{PLL})]\mathrm{d}t \\
T_{cp} &= \int_0^{t_{cp1}} [S_{cp}V_{DD} + s_{cp}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{cp})]\mathrm{d}t \\
T_{cp} &= \int_{t_{cp1}}^{t_{cp1}+t_{cp2}} [S_{cp}V_{DD} + s_{cp}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{cp})]\mathrm{d}t \\
T_d &= \int_{t_{cp1}}^{t_{cp1}+t_d} [S_d V_{DD} + s_d v_{DD}\cos(\omega_m t - \theta_0)]\mathrm{d}t
\end{aligned}
\tag{16}
$$

Here, T_{clk}, T_{cp} and T_d are the clock period, the clock path delay and the datapath delay under nominal supply voltage. This procedure is repeated numerically by sweeping θ_0 from 0 to 2π and the minimum value becomes the worst-case timing slack.

One thing to note here is that these four equations can be easily adjusted to accommodate both the phase-shifting PLL design and the phase-shifted clock distribution design. To be more specific, the impact of the phase-shifting PLL can be included by adjusting s_{PLL} and θ_{PLL} and the phase-shifted clock distribution can be represented using s_{cp} and θ_{cp}.

Chapter 4

INTRINSIC CLOCK DATA COMPENSATION

In this section, we will first verify the existence of the beneficial clock data compensate effect through HSPICE simulations in an industrial 65nm process. After that, we will examine the dependency of the clock data compensation effect on several design parameters, such as clock frequency, clock path delay and noise frequency. Modeling results on the intrinsic clock data compensation will be given at the end of this chapter.

4.1 Verification setup

In the following a few sections, we will verify the clock data compensation effect in an industrial 1.2V, 65nm process and analyze its dependency on several design parameters. The test circuit is similar to the one shown in Fig. 3 comprising a 1.9GHz clock source, an 18-stage inverter chain datapath and an 11-stage clock buffer chain with a nominal delay of 1.0ns. The delay sensitivities of the clock path and the datapath with respect to supply noise (i.e. s_{clk} and s_{data}) were both set to be 2. Here, we define delay sensitivity as the percentage increase in the path delay normalized to the percentage decrease in the supply voltage at a 10% supply noise condition. That is, a delay sensitivity of 2 means that the delay of a certain path increases by 20% for a 10% decrease in the supply voltage.

4.2 Intrinsic beneficial jitter effect

Timing slacks for different clock launching times are shown in Fig. 12 for a 200MHz resonant supply noise. The x-axis shows the time when a clock edge leaves the clock source and the y-axis shows the corresponding timing slack. The dark line represents the timing slack based on the conventional analysis which only considers the change in the datapath delay while the gray line considers the change in the clock period as well. An 11ps (or 2.1% of the clock cycle) improvement in the worst-case slack due to the beneficial jitter effect is observed.

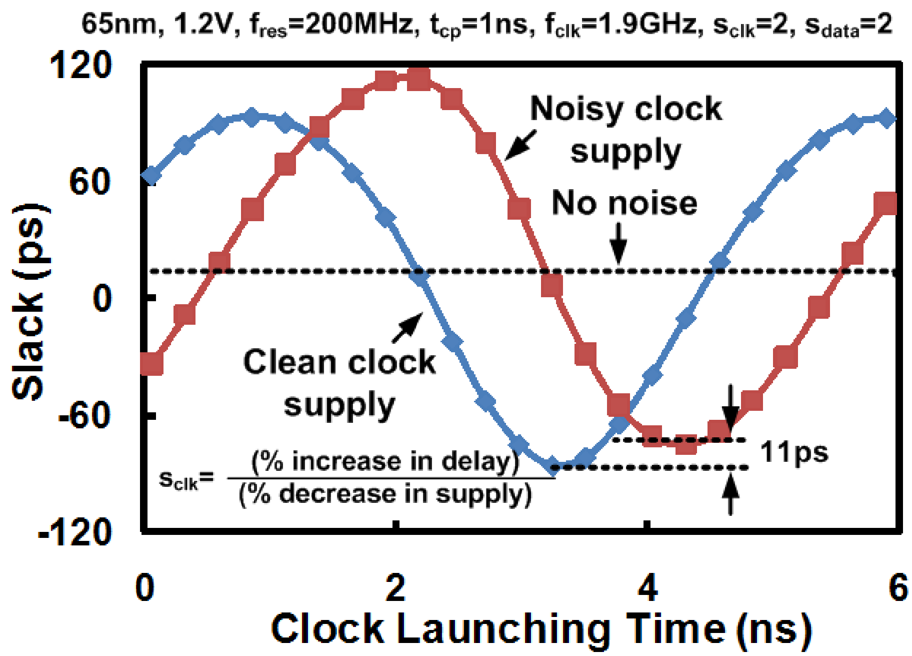


Fig. 12. Slack versus clock launching time under resonant supply noise.

4.3 Factors affecting the intrinsic beneficial jitter effect

4.3.1 Clock path delay

Fig. 13 shows the dependency of the worst-case slack on the clock path delay simulated by changing the number of clock buffer stages. For extremely long or short clock path delays, the slack considering the beneficial jitter effect (i.e. noisy clock supply) approaches the conventional analysis case (i.e. clean clock supply). This is because a very short clock path makes the clock period modulation effect weaker and conversely, a very long clock path makes each clock edge see a similar average supply voltage.

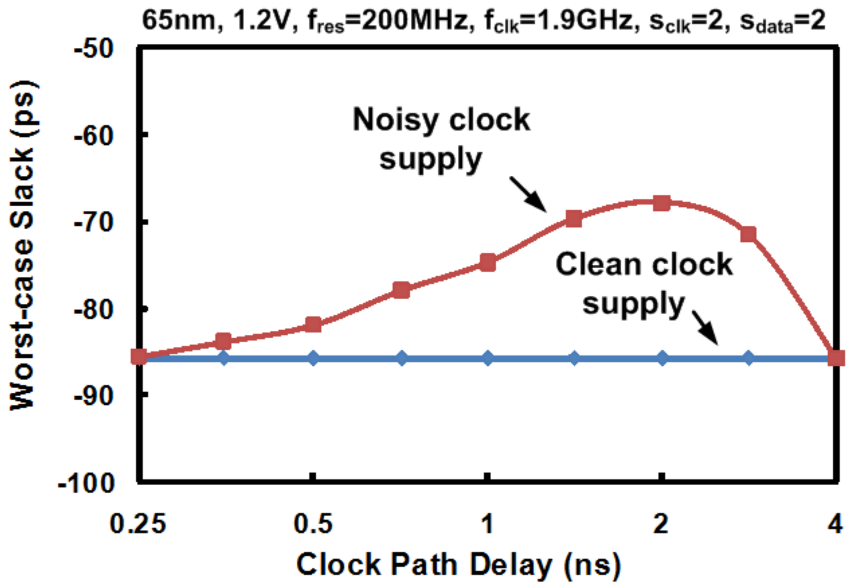


Fig. 13. Dependency of worst-case slack on clock path delay.

4.3.2 Delay sensitivity to supply noise

Fig. 14 shows the simulated worst-case slack when the datapath delay sensitivity is fixed at 2 and the clock path delay sensitivity is varied from 0 to 2.4 through the adjustment of the interconnect load, the number of clock buffer stages, and the supply noise amplitude seen by the clock path. The optimal timing compensation effect occurs when the clock path delay sensitivity is around 1.2. A clock path delay sensitivity lower than the optimal point makes the clock period less sensitive to the supply noise making the beneficial jitter effect weaker. On the other hand, a higher sensitivity eventually leads to a worse timing slack due to the excessively compressed clock periods during supply upswings.

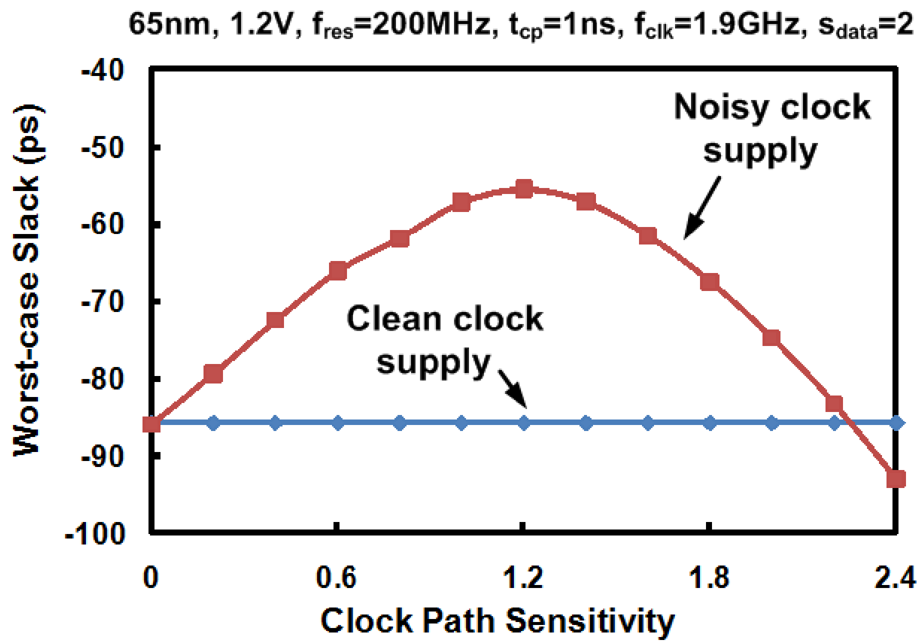


Fig. 14. Dependency of worst-case slack on clock path delay sensitivity.

4.3.3 Supply noise frequency

The worst-case slack for supply noise frequencies from 50MHz to 1.6GHz are shown in Fig. 15. At extremely low frequencies, the worst-case slack converges to the clean clock case since two consecutive clock edges see almost the same supply voltage. When the resonant frequency is high, the noisy clock supply case again converges to the clean supply case. This is because of the negligible difference in the supply voltages seen by two consecutive clock edges due to the averaging effect.

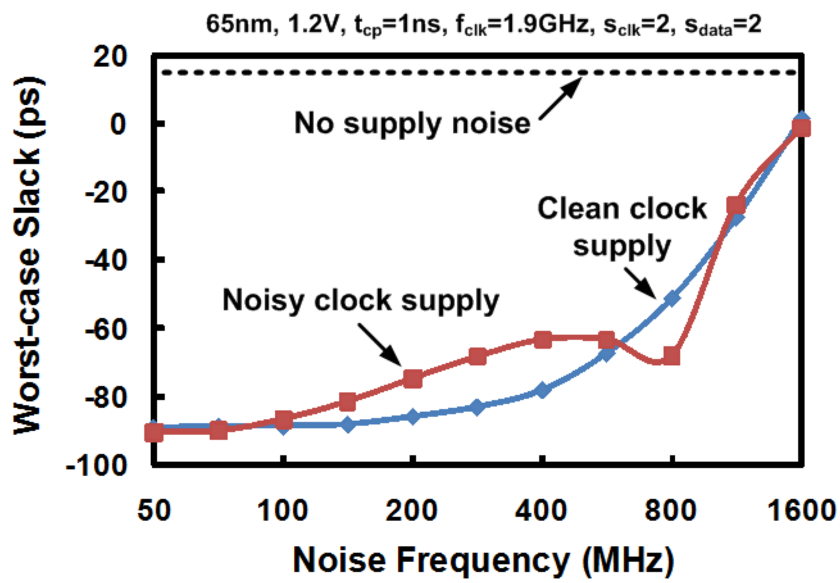


Fig. 15. Dependency of worst-case slack on supply noise frequency.

4.4 Modeling of intrinsic clock data compensation

The methodology described in Chapter 3 for modeling the beneficial jitter effect was verified with HSPICE. The clock frequency and the maximum clock skew were assumed to be 1.9GHz and 20ps, respectively [27]. A resonant noise with a frequency of 200MHz and an amplitude of 10%*V_{dd} was used for the simulations.

In the first test, setup and hold time margins were examined for different clock path delays. The results in Fig. 16 show a 45ps change in the setup time margin and the detailed behavior is precisely captured by the proposed model. When compared with previous models, the maximum estimation error is improved from 26ps to only 3ps. Moreover, our proposed model also closely matches the simulation results for hold time margin as shown in Fig. 17. The maximum error is less than 1ps for all clock path delays used in the simulations. A latch-based pipeline circuit was also simulated and the results are summarized in Table 1.

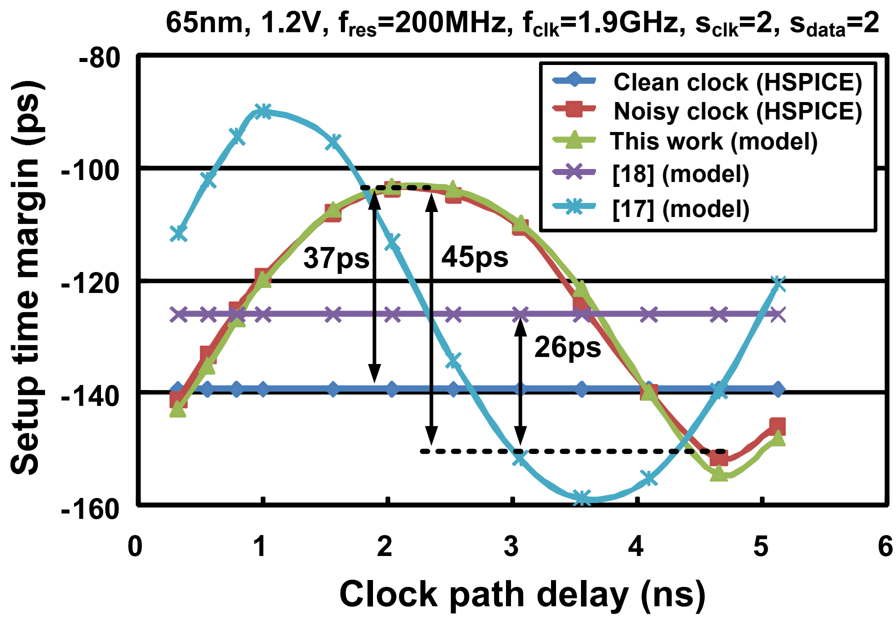


Fig. 16. Dependency of setup time margin on clock path delay.

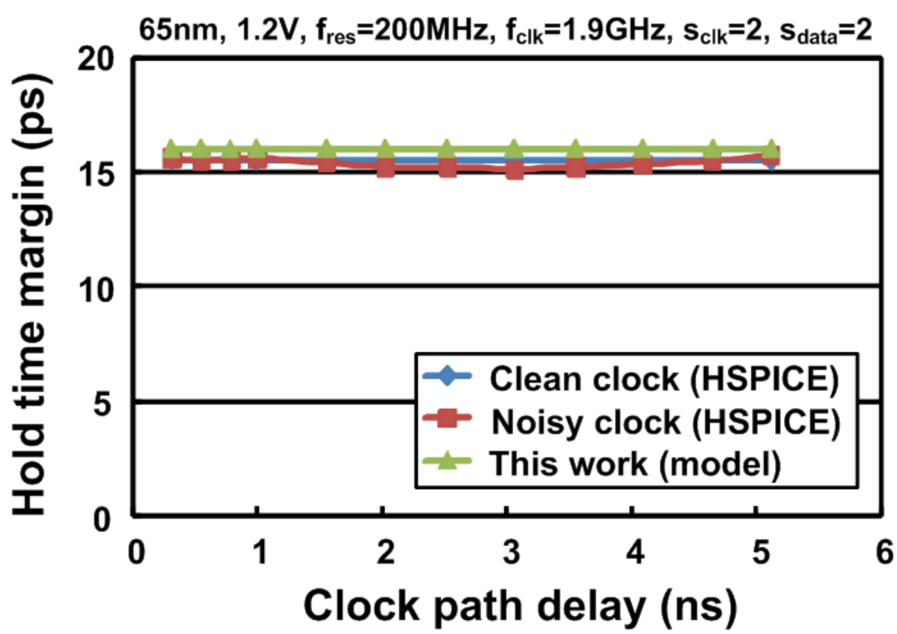


Fig. 17. Dependency of hold time margin on clock path delay.

Table 1. Maximum modeling error for different clock path delays (f_{clk}=1.9GHz, f_{res}=200MHz, s_{clk}=2, s_{data}=2)

	Register-based		Latch-based	
	Setup	**Hold**	**Setup**	**Hold**
[17]	41ps	N/A	37ps	N/A
[23]	26ps	N/A	32ps	N/A
This work	3ps	1ps	7ps	1ps

We also tested the accuracy of the model for different supply noise frequencies. As shown in Fig. 18, the setup time margin is improved due to the beneficial jitter effect for a typical resonant frequency range of 100MHz to 300MHz. Similar to the previous test, both setup and hold time margins were simulated for register-based and latch-based pipeline circuits and the results are summarized in Table 2. A significant improvement in the modeling accuracy is achieved.

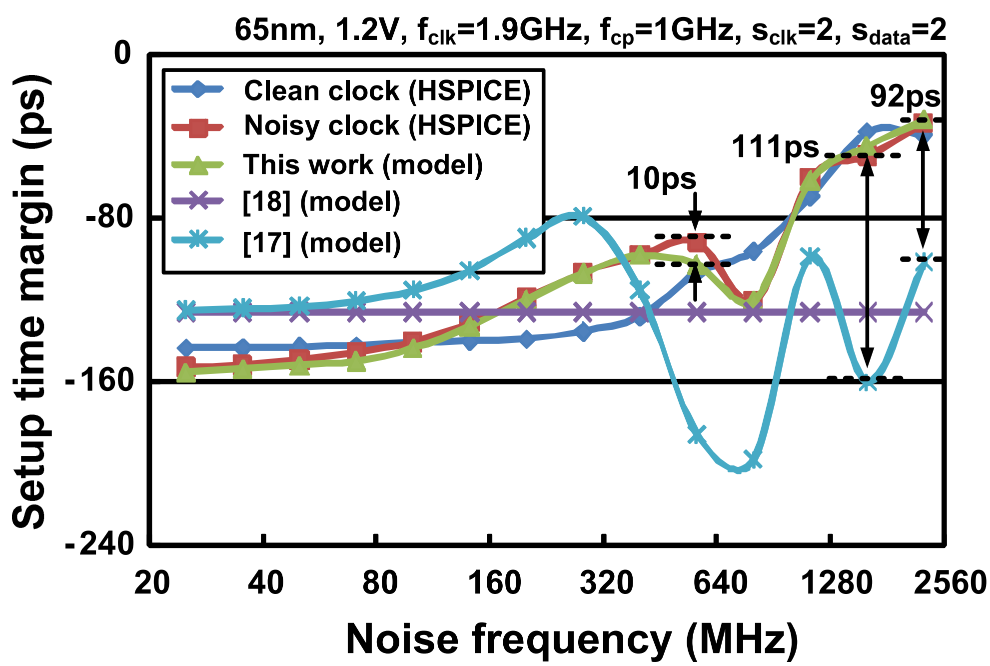


Fig. 18. Dependency of setup time margin on supply noise frequency.

Table 2. Maximum modeling error for different noise frequencies (f_{clk}=1.9GHz, t_{cp}=1ns, s_{clk}=2, s_{data}=2)

	Register-based		**Latch-based**	
	Setup	**Hold**	**Setup**	**Hold**
[17]	111ps	N/A	105ps	N/A
[23]	92ps	N/A	96ps	N/A
This work	10ps	1ps	10ps	1ps

Chapter 5

PHASE-SHIFTED CLOCK DISTRIBUTION

In this section, we will propose a phase-shifted clock distribution design which could modulate the clock period in order to enhance the clock data compensation effect. An adaptive phase-shifting PLL will also be proposed in this section with extensive measurement results from a 65nm test chip validating its performance. We will provide the simulation results of the proposed PLL in a 32nm process and discussions on a few design considerations at the end of this section.

5.1 Phase-shifted clock buffer designs

The clock data compensation effect in its intrinsic form provides modest timing margin relief for pipeline circuits. This is because the point when the clock period is stretched out the most (i.e. point “A” in Fig. 19) does not coincide with the point when the delay is the longest (i.e. point “B” in Fig. 19). It is important to note that the former situation occurs when the supply voltage has a negative slope while the later occurs when the instantaneous supply voltage is the lowest. In order to maximize the timing compensation effect, the phase of the supply noise seen by the clock path should be shifted such that points A and B are aligned.

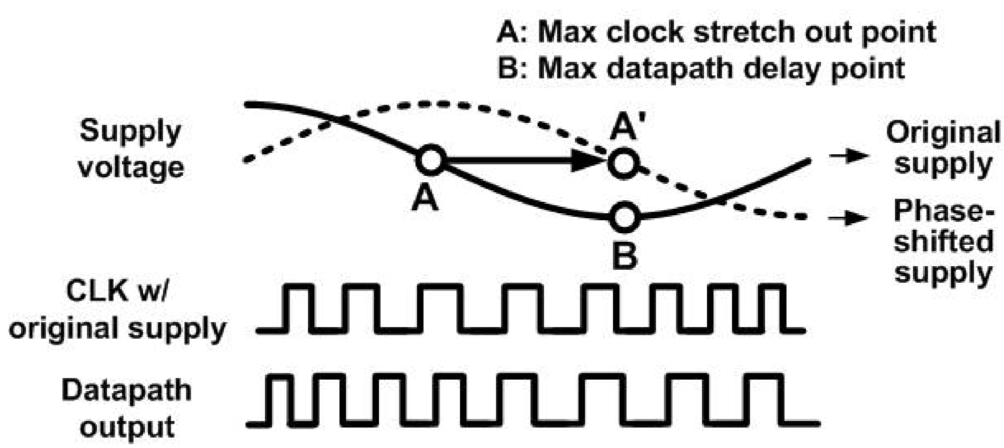


Fig. 19. Concept of the phase-shifted clock buffer design.

Fig. 20(left) shows the schematic of a conventional buffer and various phase-shifted clock buffers for enhancing the beneficial effect [22][24]. The previous RC filtered buffer contains a PMOS pull-up device and an NMOS capacitor to generate a phase-shifted supply. The main drawback of this design is the large area. The resistance of the RC filter must be very small to minimize the IR drop (e.g. 50mV or less) which in turn requires a large capacitance to obtain the desired supply phase shift. As shown in Fig. 20(right), the layout area of the RC filtered buffer is about 10× larger than that of a conventional buffer.

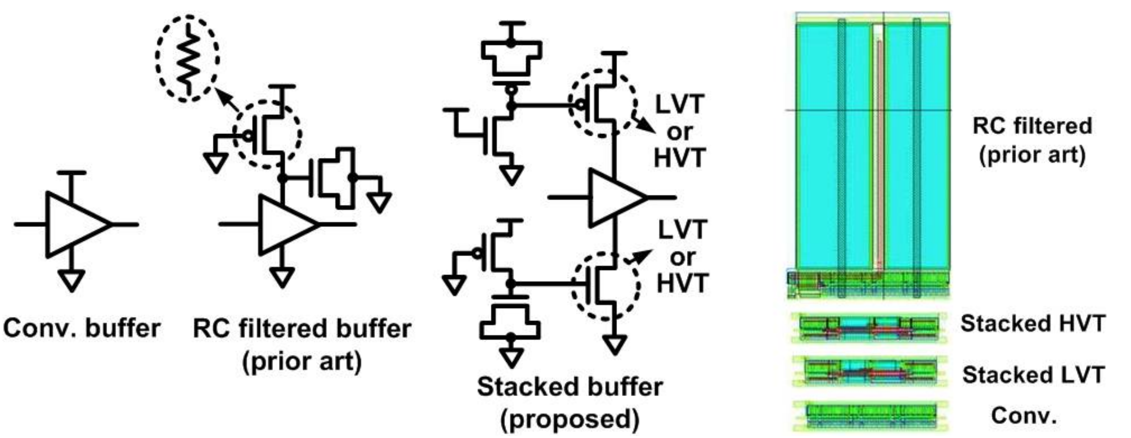


Fig. 20. (left) Schematic of a conventional buffer, an RC filtered buffer, and the proposed stacked high V_t and low V_t buffers. (right) Layout of the different clock buffers.

Based on those observations, we propose a phase-shifted clock buffer using stacked devices to significantly reduce the buffer area while achieving a similar timing improvement. Fig. 20 shows the schematic and layout of the new circuit where header and footer devices controlled by separate RC filters are used instead of an explicit RC filter for generating a phase shifted supply. MOSFETs operating in the linear mode are used for implementing the resistors, enabling a much smaller layout area. The beneficial jitter effect can be further enhanced by using high V_t header/footer devices to make the buffer delay more sensitive to the phase-shifted supply noise. Hence, the proposed stacked buffer design was evaluated for both low V_t (LVT) and high V_t (HVT) header and footer devices. Since the actual switching current no longer flows through the resistor in the new design, small devices with large resistances can be safely used for the RC filter which in turn reduces the capacitor area for achieving the desired phase shift. As shown in Fig. 20(right), the layout area of the proposed buffer is only 10% of the previous RC filtered buffer area. Even after considering the fact that the proposed stacked buffer has to be 50% larger than the conventional buffer for the same drive current, an 85% saving in buffer layout area can be achieved.

Table 3. Power consumption of different clock buffer designs (f_{clk}=1.9GHz)

	Conv.	**RC Filtered (prior art)**	**Stacked (this work)**
Clean V_{dd}	5.013mW	4.868mW	4.922mW
Noisy V_{dd}	5.116mW	5.493mW	5.024mW

Power consumption is another major consideration for clock network designs. Table 3 compares the power consumption of a representative 9-stage clock path using the three different clock buffers. Simulation results show that both phase shifted designs consume slightly less power than the conventional buffer in case of no supply noise (i.e. clean V_{dd}). This is because the header/footer devices reduce the effective supply voltage seen by the buffer which reduces the CV^2 and short circuit power dissipation. Applying a 120MHz resonant noise to the supply voltage (i.e. noisy V_{dd} case, the noise amplitude is 10% of the nominal supply voltage) led to a 12.8% increase in power consumption for the

RC filtered buffer due to the power wasted for charging and discharging the large capacitor. In contrast, the proposed stacked buffer design shows only a 2.1% power increase owing to the greatly reduced capacitor size.

5.2 Modeling of phase-shifted clock distribution

Our proposed model can be applied to the phase-shifted clock distribution design by introducing a parameter φ which indicates the amount of supply noise phase shift. More specifically, when solving for t_{cp1} and t_{cp2} in (6), we use the following expression for the propagating velocity:

$$v(t) = SA_0 + sa\cos\varphi\cos(\omega_m t - \theta - \varphi) \tag{17}$$

HSPICE simulations were performed for the phase-shifted clock distribution to evaluate the accuracy of the proposed model. The test circuit is similar to the one shown in Fig. 3 with RC filtered buffers used in the clock network. The value of R is chosen to be as large as possible while satisfying the IR drop requirement of less than 50mV. Fig. 17 shows the setup time margin for different phase shift values. An optimal phase shift value makes the maximum clock period point coincide with the maximum datapath delay point. Simulation results and the estimated values using different models are given in Fig. 21, from which we can see that our proposed model reduces the maximum estimation error from 22ps to 6ps. The hold time margin was also simulated for a phase shift value of 0.2π which gives the best setup time margin. The maximum modeling error for this configuration was only 4ps.

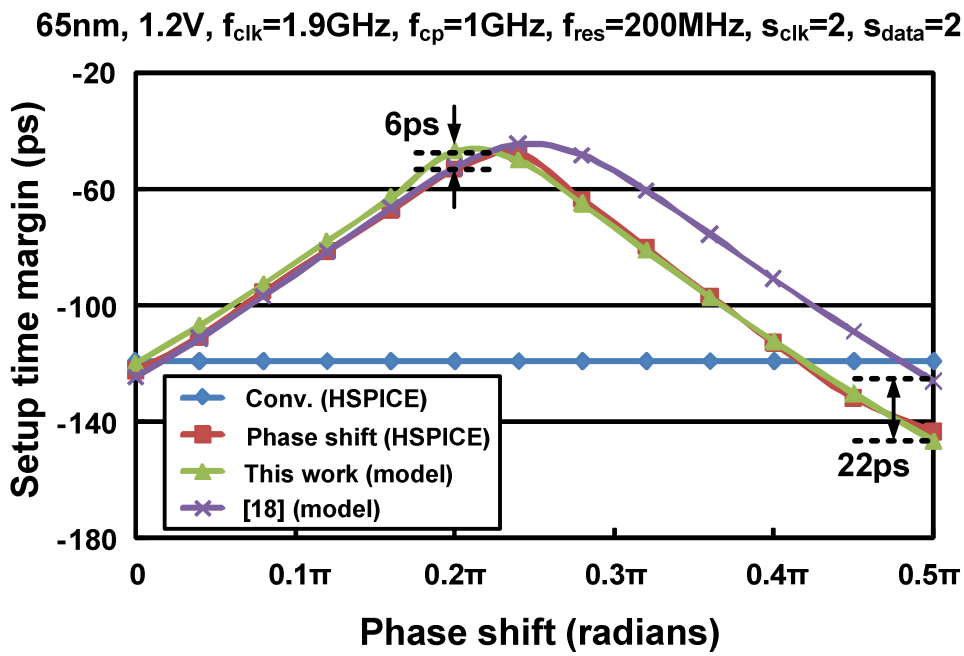


Fig. 21. Dependency of setup time margin on phase shift.

5.3 Test chip organization

A 65nm test chip was designed to verify the performance of the proposed phase-shifted clock buffers. Fig. 22 shows the block diagram of the proposed test chip which contains two VCOs, a clock path block, a core logic block, two 13-bit counters, a noise injection block, a supply noise sensor, and a read-out block. Two starved ring oscillator based VCOs are used to generate the clock signal and the supply noise. By adjusting the external bias voltage VBIAS, the VCO frequency can be raised up to 3.4GHz. Five clock paths are implemented with different clock buffers: the conventional buffer, the RC filtered buffer, the stacked LVT buffer, the stacked HVT buffer and a “no buffer” design in which the output of the clock VCO is directly connected to the local registers. Each path contains 9 buffer stages and long interconnects giving a clock path delay of 1.0ns.

One clock path is selected at a time to test each clock buffer design separately. The datapath circuit consists of two standard D-flip-flops and a ten-stage FO4 inverter chain in between to represent a critical path with a nominal delay of 0.6ns. Input to the datapath is toggled between 1 and 0 in each cycle. Additional control logic increments the "data counter" only when the sampled output and the corresponding input are identical (during input '1' cycles only). A "reference counter" increments every other cycle, and is used for counting the total number of sampled outputs. By scanning out the number stored in the data counter when the reference counter overflows, the percentage of correct samples can be conveniently measured. The noise injection block has 32 NMOS devices that can be clocked by the noise VCO. By adjusting the noise VCO frequency and activating different number of noise injection devices, the desired noise current can be injected into the supply network. A supply sensor is also designed for on-chip noise measurements. This circuit receives the noisy supply and ground signals as differential inputs, and the output indicates the supply noise frequency and amplitude [13]. The read-out block consists of a 10-bit parallel-to-serial shift register and additional control logic. In COUNT mode, the shift register captures the upper 10 bits of the data counter whenever the reference counter overflows. In READ mode, an external clock is provided to scan out the stored data serially. Fig. 23 shows the read-out waveforms including a mode selection signal, an external clock, and a read-out scan value. The read-out value we record is the average of 512 scan values to eliminate transient noise effects.

Note that a VCO-controlled noise injection block generates supply noise at a specific frequency (plus harmonics) making it easier to characterize the various clock buffers at a given noise frequency. As explained in the introduction section, supply noise at the resonant frequency has been shown to be the dominant component in high performance microprocessors so the global supply noise generated by a VCO-based noise injection block is a simple yet effective way of generating a representative supply noise. Of course, one can consider using more elaborate digital blocks for generating global and local supply noises but the drawback here is that it may be difficult to know the exact supply noise waveform used for the chip testing.

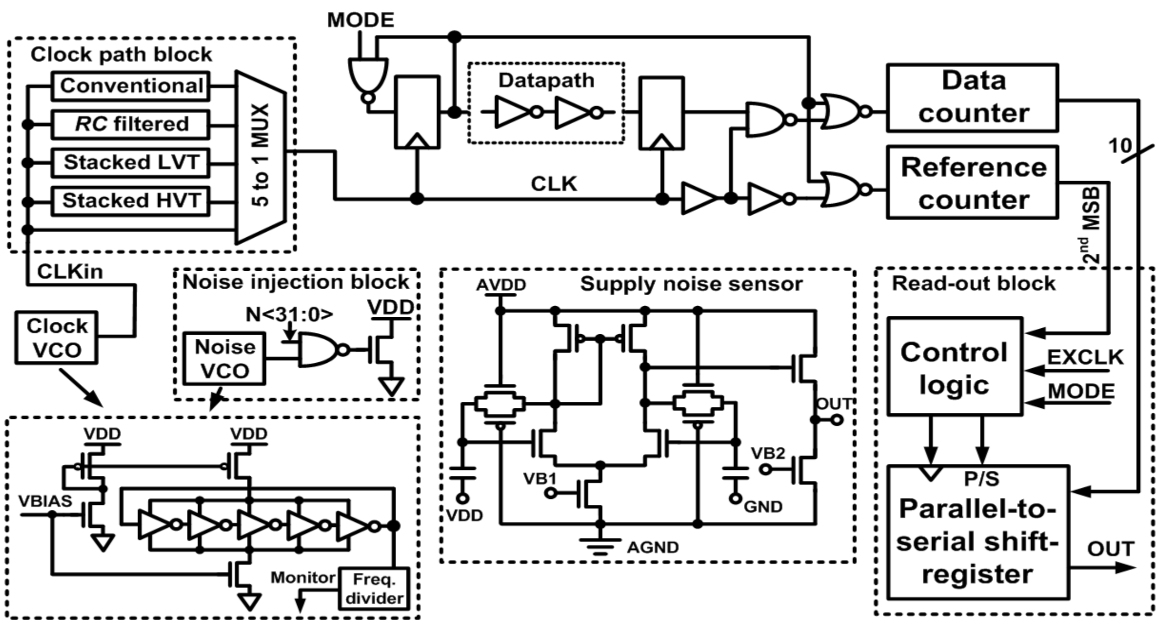


Fig. 22. High level block diagram of the 65nm test chip.

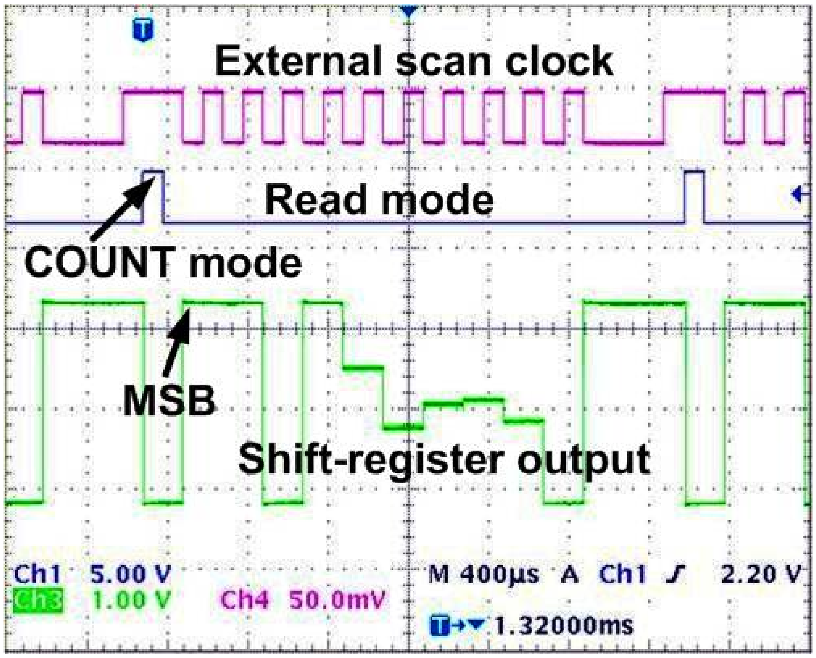


Fig. 23. Example read-out waveforms from the 65nm test chip.

5.4 Test chip measurement results

The test chip was fabricated in a 1.2V, 65nm Low Power (LP) process and the die photo is shown in Fig. 24. In the first test, eight noise injection devices were turned on and the noise VCO bias was adjusted to generate a 118MHz noise which corresponds to the resonant frequency of the fabricated test chip. Fig. 25 shows the percentage of correct samples measured from the different clock paths. F_{max} or the maximum operating frequency is defined as the frequency at which the percentage of correct samples starts to drop. F_{max} of the conventional buffer design reduced from 1.64GHz to 1.2GHz when the supply noise injection circuit was activated. F_{max} of the RC filtered buffer, the stacked LVT and HVT buffers were 1.33GHz, 1.31GHz and 1.34GHz, respectively, which translate into roughly a 10% performance improvement compared with a conventional buffer design.

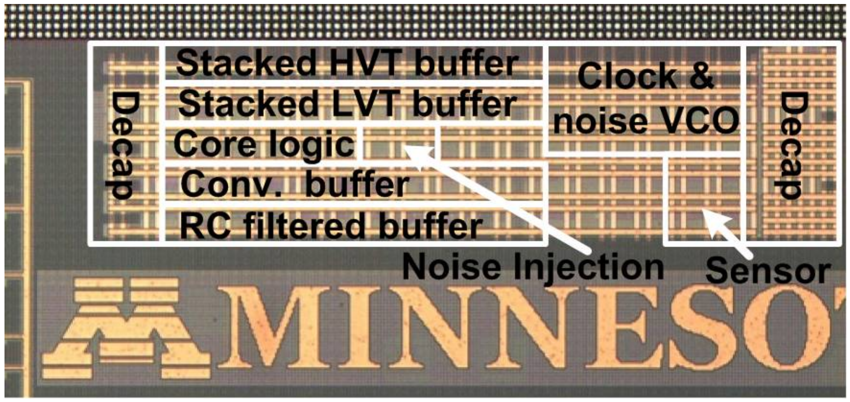


Fig. 24. Chip microphotograph and floor plan.

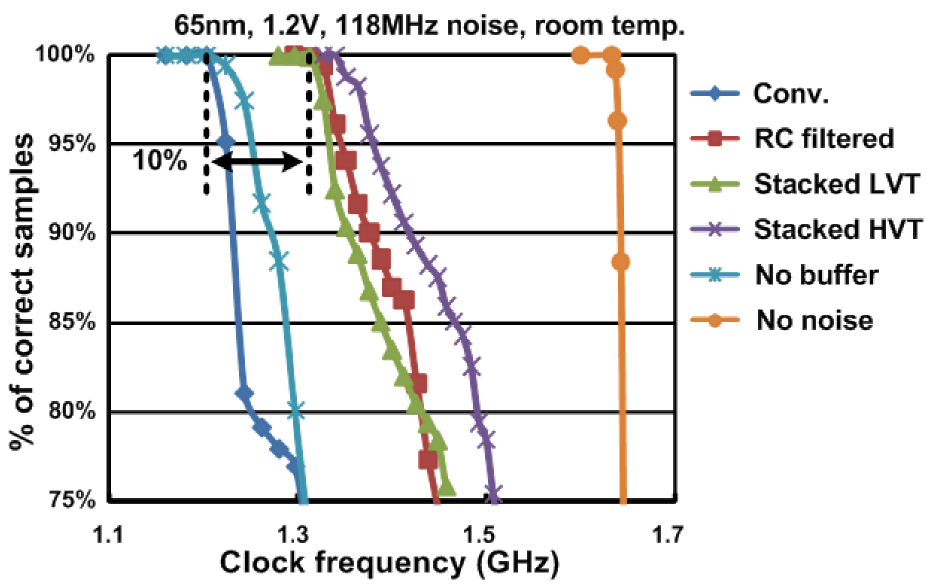


Fig. 25. Measured bit error rate for different clock buffer designs.

Fig. 26 shows the measured F_{max} for the different clock buffer designs when increasing the number of noise injection devices. The supply noise frequency is maintained at

118MHz. As expected, the maximum frequency decreases linearly with more number of noise injection devices turned on. The proposed stacked buffer designs improve the F_{max} by 8-15% when more than 8 noise injection devices are turned on. This is similar to what the RC filtered buffer design achieves under the same condition.

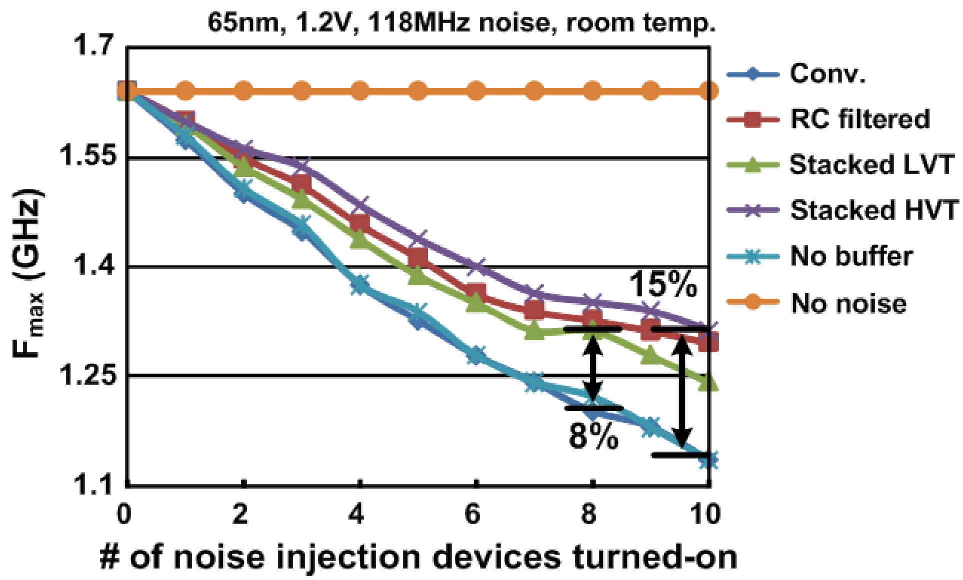


Fig. 26. Measured F_{max} for different number of noise injection devices.

The normalized F_{max} of the different designs are shown in Fig. 27 for a noise frequency range between 10MHz and 1.2GHz. The number of noise injection devices is carefully adjusted so that F_{max} of the conventional buffer design is fixed at 1.2GHz. The figure shows that F_{max} of the phase-shifted clock buffer designs is improved by 8-27% for a typical resonant frequency range of 100MHz to 300MHz. For noise frequencies higher than 400MHz or lower than 50MHz, F_{max} of the phase-shifted clock buffer designs and the conventional design are similar. This is because the clock cycle modulation effect is very weak in both extreme frequency cases as explained in Section III.2: when the noise frequency is high, the strong averaging effect makes consecutive clock edges see almost the same average supply voltages; when the noise frequency is low, consecutive clock edges again see almost the same supply voltages since it fluctuates very slowly. At some high frequencies, the phase-shifted buffer designs exhibit some performance degradation but this does not affect the overall performance because the worst-case noise scenario always happens in the resonant band, rather than at higher frequencies [21].

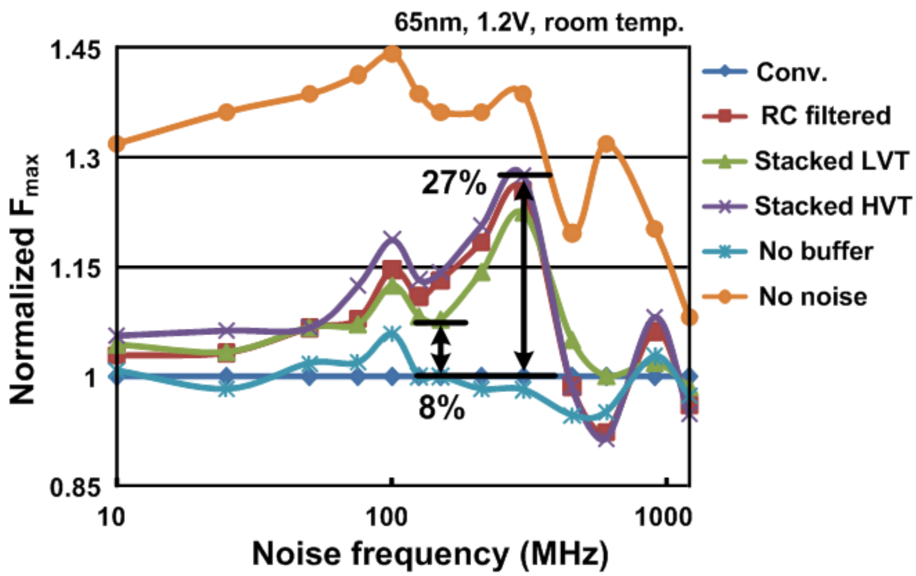


Fig. 27. Measured F_{max} normalized to the conventional buffer case for different noise frequencies.

5.5 Comparison with the adaptive clock scheme

An alternative way of enhancing the beneficial jitter effect is to modulate the clock period at the clock source (e.g. PLL) so that the clock period stretching effect is maximized by the time the clock signal arrives at the flip-flops. Adaptive clocking schemes based on this principle have been recently deployed in Intel Nehalem processors [15]. In this scheme, the clock frequency of the PLL output is carefully designed to track the supply voltage variation with a phase difference as shown in Fig. 28. The proposed phase-shifted clock buffer design can be used in conjunction with existing adaptive clocking schemes to further improve chip performance. The effectiveness of using both techniques in tandem for improving chip performance was verified with the test circuit shown in Fig. 29. The VCO output frequency was designed to follow the supply noise with a certain phase shift and a noisy power supply was applied to all blocks. The noise amplitude was set to be 10% of the nominal supply voltage. The simulated timing slack is shown in Fig. 30 for a noise frequency range from 10MHz to 1.2GHz. It is shown that the adaptive clocking scheme alone achieves a 17-39ps worst-case slack improvement for a typical resonant frequency range between 100MHz and 300MHz. The phase-shifted buffer scheme provides an additional 30-62ps improvement in timing slack.

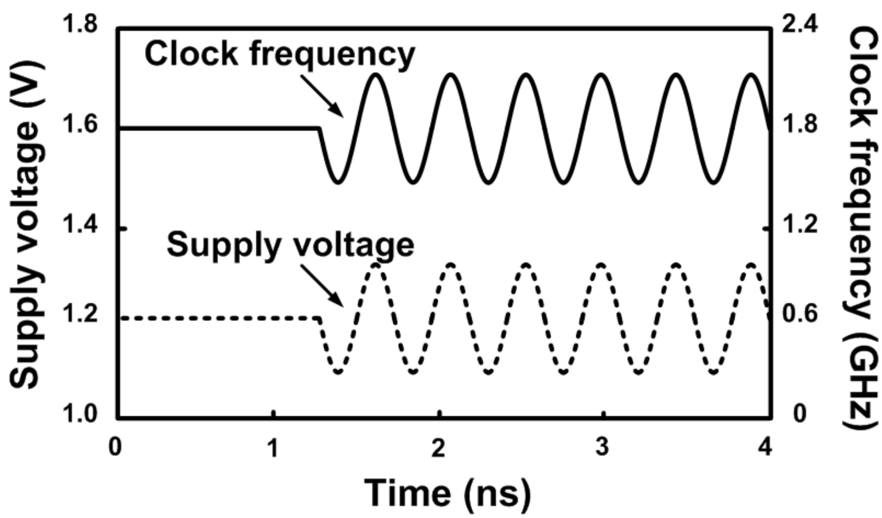


Fig. 28. The PLL output frequency is modulated by the supply noise in adaptive clocking schemes.

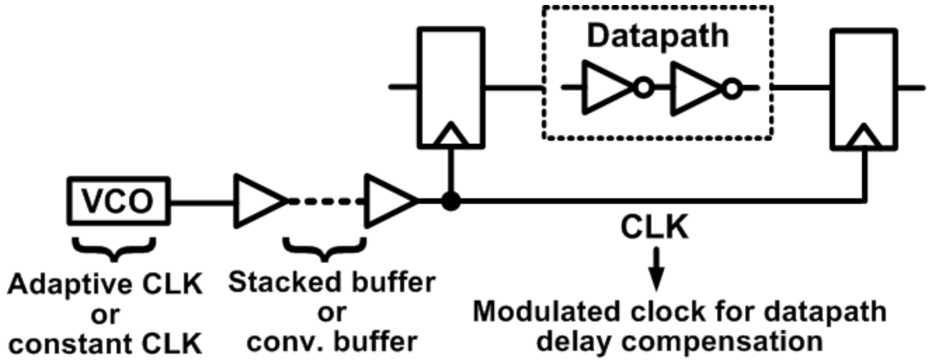


Fig. 29. Clock cycle modulation schemes.

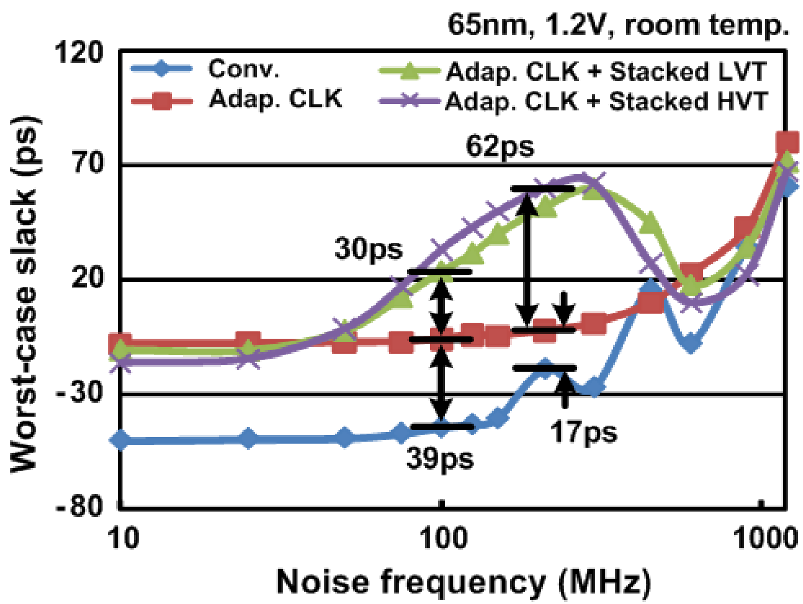


Fig. 30. Simulated worst-case slack for different clock cycle modulation schemes.

The setup and hold time margins of the adaptive clocking scheme can be mathematically derived through the following steps. Assume that the supply voltage is expressed as

$$V_{dd}(t) = V_{dd0} + v_{dd}\cos(\omega_m t) \tag{18}$$

where V_{dd0} and v_{dd} are the DC and AC amplitudes and ω_m is the supply noise frequency. We can expect the clock frequency f_{clk} of this PLL to vary at the same frequency, i.e., f_{clk} can be written as

$$f_{clk}(t) = f_{clk0} + f_{ac}\cos(\omega_m t - \varphi)\,. \tag{19}$$

Here, f_{clk0} and f_{ac} are the DC and AC amplitude and φ denotes the phase shift between the supply noise and the frequency variation. We apply our proposed model to the adaptive clocking scheme by varying t_{clk1} in (6) depending on the time when the first clock edge is triggered, emulating the behavior of the adaptive clock frequency. The detail expression of t_{clk1} is determined by (19). To corroborate the model, we ran simulations using the circuit given in Fig. 2 with a conventional clock path and a supply-tracking PLL. φ in (19) was swept from -π to π and f_{ac} was swept from $0.12f_{clk0}$ to $0.32f_{clk0}$. Simulation results in Fig. 31 show that the optimal setup time margin is achieved when φ is 0 and f_{ac} is $0.2f_{clk0}$. The estimation error of the timing model is only 6ps.

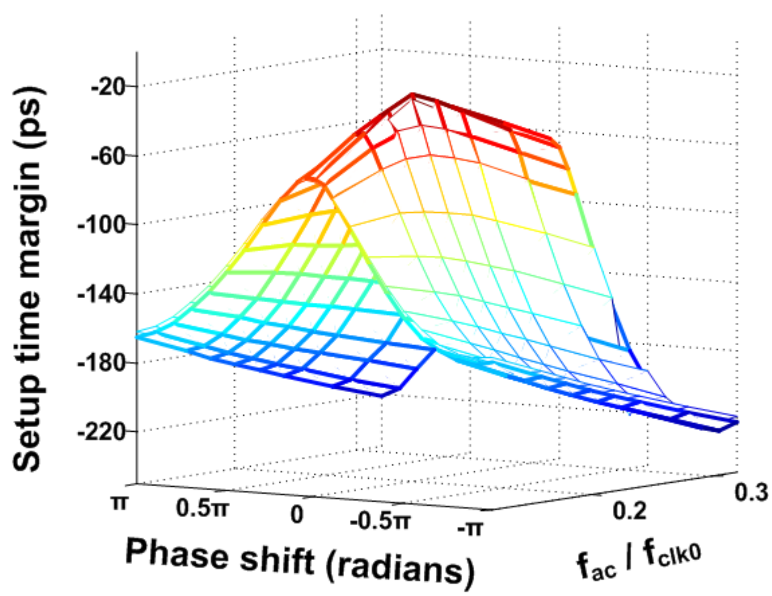


Fig. 31. Setup time margin versus design parameters of clock cycle modulation schemes.

5.6 Partially phase-shifted clock distribution design

Since the phase-shifted clock buffers are larger than (or have lower drive current than) conventional buffers, a more economical approach would be to limit their use to global clock buffer stages. We refer to this implementation as the "partially phase-shifted design" which is illustrated in Fig. 32. Simulation results of the worst-case slack are shown in Fig. 33 for different numbers of global clock buffer stages using the stacked LVT buffers. Since the number of buffers at each clock hierarchy increases exponentially in an H-tree type topology, the area overhead can be significantly reduced by using conventional buffers in the final stages of the clock network. As shown in Fig. 33, using phase-shifted clock buffers in the first 9 out of 11 stages in the clock network can provide a 52ps improvement in the worst-case slack (about 71% of the maximal possible improvement) while reducing the clock buffer area overhead by 75%.

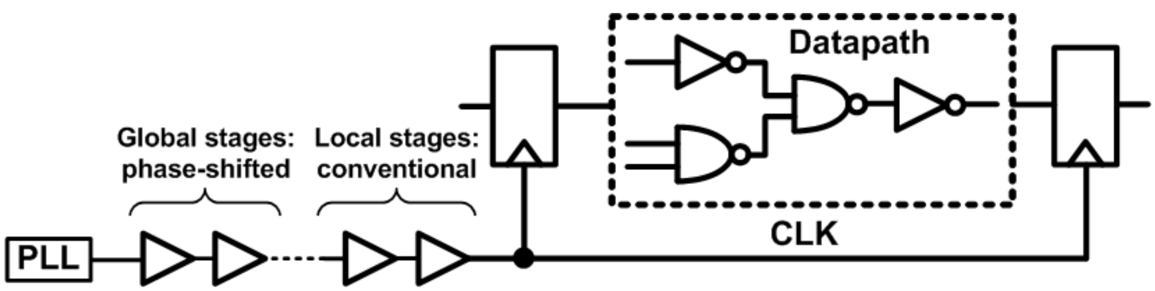


Fig. 32. Partially phase-shifted clock distribution design.

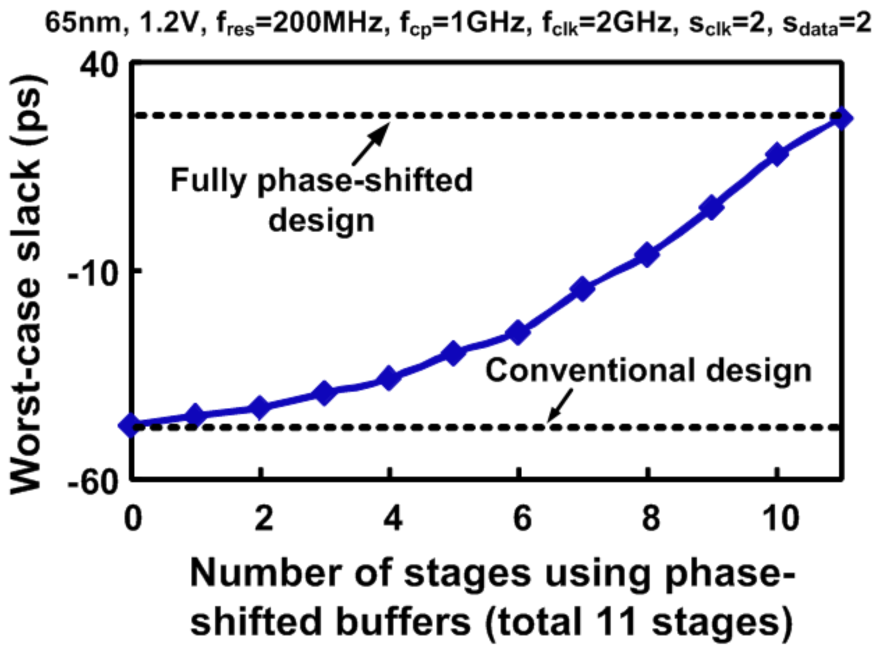


Fig. 33. Slack improvement using a partially phase-shifted clock distribution design.

5.7 Impact of PVT variations

Most of the analysis in the previous sections assumes that the clock path and datapath have the same delay sensitivities. In reality, the delay sensitivity may vary depending on the amount of interconnect. For example, a clock path may have a lower sensitivity because of its long interconnect, and a datapath may also have a low sensitivity if it is wire dominated, like in data buses. To verify the performance of the phase-shifted clock distribution technique for different delay sensitivities, we present simulation results of the worst-case slack in Fig. 34 where the delay sensitivity of the datapath is fixed at 2 while the delay sensitivity of the clock path is swept from 1.6 to 2.4. The figure clearly shows that the worst-case slack is improved using the proposed clock buffer for the entire delay sensitivity range. Fig. 34 also shows the average and 3σ values of the worst-case slack from Monte Carlo simulations with random local t_{ox} and V_t variations. Despite the slight degradation in the timing slack, the proposed stacked clock buffer design provides a consistent timing improvement in the presence of random process variation at 25ºC and 110ºC.

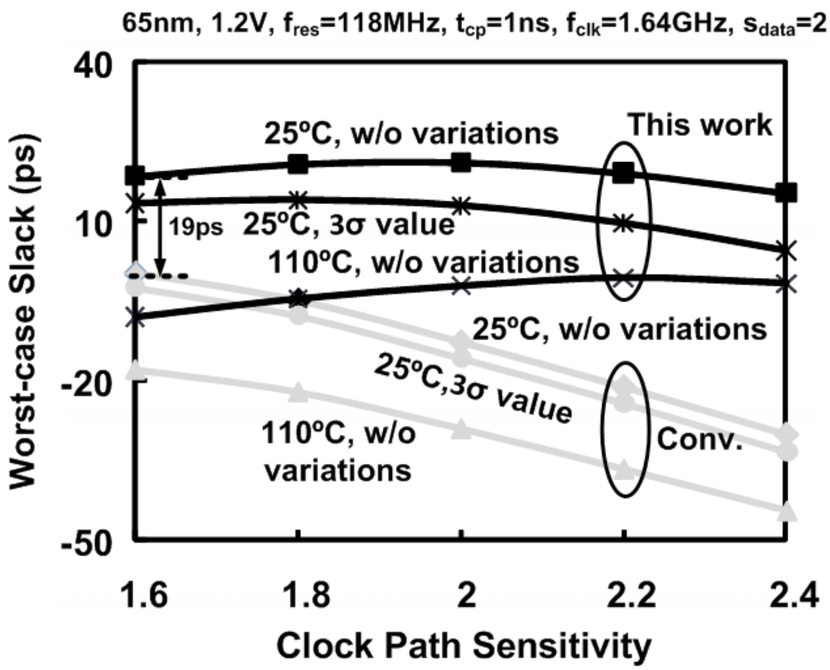


Fig. 34. Impact of random process variation on the worst-case slack at 25ºC and 110ºC. Monte Carlo simulations were performed using the following parameters: $V_{t,N}$: σ/μ=3.6%, $V_{t,P}$: σ/μ=1.6%, $t_{ox,N}$: σ/μ=0.6%, $t_{ox,P}$: σ/μ=0.6%.

Chapter 6

ADAPTIVE PHASE-SHIFTING PLL

In this section, we will briefly review the existing models for clock data compensation effect and use the numerical model to analyze the clock data compensation effect and the adaptive clocking schemes. An adaptive phase-shifting PLL will also be proposed in this section with extensive measurement results from a 65nm test chip validating its performance. We will provide the simulation results of the proposed PLL in a 32nm process and discussions on a few design considerations at the end of this section.

6.1 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 that whether the existing designs could achieve the optimum compensation. To answer this question, let us first have a brief analysis of the adaptive clocking scheme as shown in Fig. 31. The four waveforms represent the supply voltage with resonant noise and the clock period modulation effect seen by the PLL, the clock distribution and the local registers, respectively. The minimum supply voltage occurs at point “A”, which is also the point when the datapath delay is worst. Suppose the adaptive PLL produces the longest clock period at “B” [25] 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, local registers see the maximum clock cycle at time

“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, a certain amount of 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 “B_{opt}”. Previous designs, however, did not consider both effects and were not able to adapt to different design parameters. Motivated 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 optimum performance.

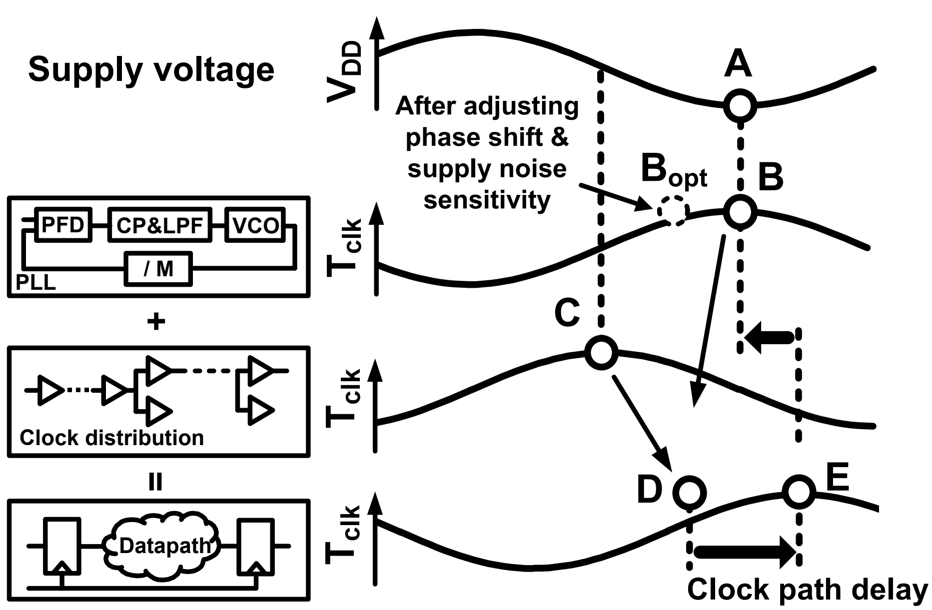


Fig. 35. Illustration of adaptive clocking schemes for clock data timing compensation.

6.2 Modeling of adaptive clocking schemes

Next we will use a standard register-based pipeline circuit shown in Fig. 3 to describe the flow 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 arrive at the register. The input data of the first register starts to propagate through the datapath at time t=t_{cp1} and takes t_d to reach the input of the second register. 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

$$slack = t_{clk} + t_{cp2} - t_{cp1} - t_d \tag{20}$$

Similar to (3), four equations can be established for t_{clk}, t_{cp2}, t_{cp1} and t_d as follows:

$$\begin{aligned}
T_{clk} &= \int_0^{t_{clk}} [S_{PLL}V_{DD} + s_{PLL}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{PLL})]\mathrm{d}t \\
T_{cp} &= \int_0^{t_{cp1}} [S_{cp}V_{DD} + s_{cp}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{cp})]\mathrm{d}t \\
T_{cp} &= \int_{t_{cp1}}^{t_{cp1}+t_{cp2}} [S_{cp}V_{DD} + s_{cp}v_{DD}\cos(\omega_m t - \theta_0 - \theta_{cp})]\mathrm{d}t \\
T_d &= \int_{t_{cp1}}^{t_{cp1}+t_d} [S_d V_{DD} + s_d v_{DD}\cos(\omega_m t - \theta_0)]\mathrm{d}t
\end{aligned} \tag{21}$$

Here, T_{clk}, T_{cp} and T_d are the clock period, the clock path delay and the datapath delay under nominal supply voltage. This procedure is repeated numerically by sweeping θ_0 from 0 to 2π and the minimum value becomes the worst-case timing slack.

One thing to note here is that these four equations can be easily adjusted to accommodate both the phase-shifting PLL design and the phase-shifted clock distribution design. To be more specific, the impact of the phase-shifting PLL can be included by adjusting s_{PLL} and θ_{PLL} and the phase-shifted clock distribution can be represented using s_{cp} and θ_{cp}.

As it has been 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 optimum clock data compensation. In this section, we will apply the numerical model to a standard pipeline circuit to provide a deeper insight to the adaptive clocking schemes. The clock path delay of the circuit under test is 1.0ns and the clock period and datapath delay under nominal supply voltage are both 0.83ns. Fig. 36 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 to 150MHz and the clock distribution under test includes a large RC filter which reduces the supply noise seen by the clock buffers by 80% [23]. Accordingly, s_{cp} and θ_{cp} are set to $0.2s_d$ and 0.435π in the numerical model to account for the impact of this phase-shifted clock distribution design. As shown in fig. 7(left), the optimum slack can be achieved when s_{cp}=$1.0s_d$ and θ_{cp}=0.3π. In the second test, the resonant noise is set to 40MHz and the clock distribution under test is assumed to be a chain of inverters with long interconnect in between. Therefore, s_{cp} and θ_{cp} are set to $0.7s_d$ [22] and 0, respectively. Simulation results of the worst-case slack are provided in Fig. 36(right) showing an optimum configuration at s_{PLL}=$1.05s_d$ and θ_{PLL}=0.05π. As it can be seen from Fig. 36, the optimum configuration can vary a lot depending on the clock distribution design, resonant frequency, etc. These results again confirmed the need of programmability on phase shift and supply noise sensitivity in order to achieve the optimum performance under different operating conditions.

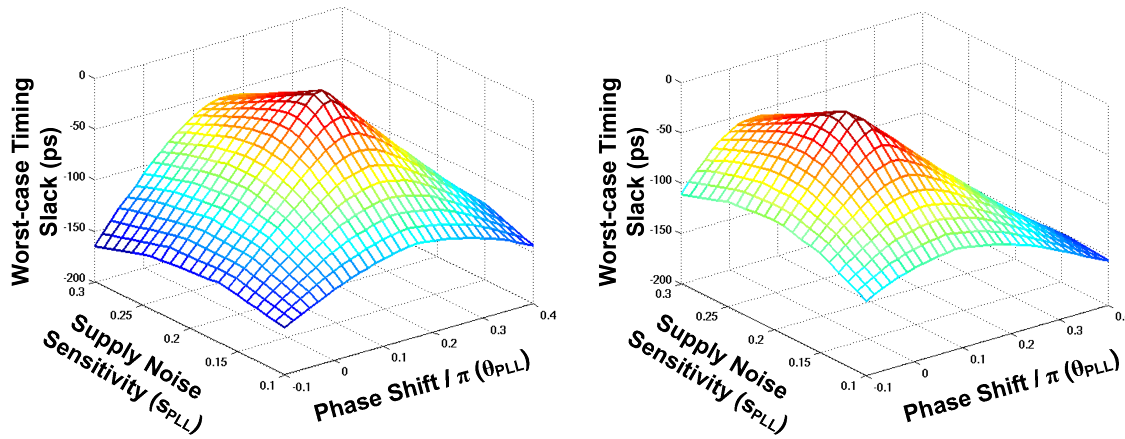


Fig. 36. Dependency of the worst-case slack on phase shift (θ_{PLL}) and supply noise sensitivity (s_{PLL})

The numerical model has also been applied to several other clock distribution designs with different characteristics, i.e., different θ_{cp} and s_{cp}, and the results are summarized in Table 4. As shown in this table, the optimum configuration, i.e., θ_{PLL} and s_{PLL}, of the adaptive phase-shifting PLL design can vary a lot depending on the clock distribution characteristics. It is interesting to look into an extreme case when there is no supply noise in the clock distribution (clock tree #4). As it can be expected, the maximum clock period point needs to be shifted by 1ns (clock path delay) so that it could compensate the maximum datapath delay point. Since the noise frequency is 80MHz, 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 the clock trees having the same supply noise sensitivity as the datapath. As it can be seen from the modeling results for clock tree #5, #6 and #7, no phase shift is needed for different resonant frequencies. We can also see that by choosing the optimum configuration for the proposed PLL, the worst-case timing slack can be improved by 42- 201ps, which is equivalent to 5- 24% of the clock period.

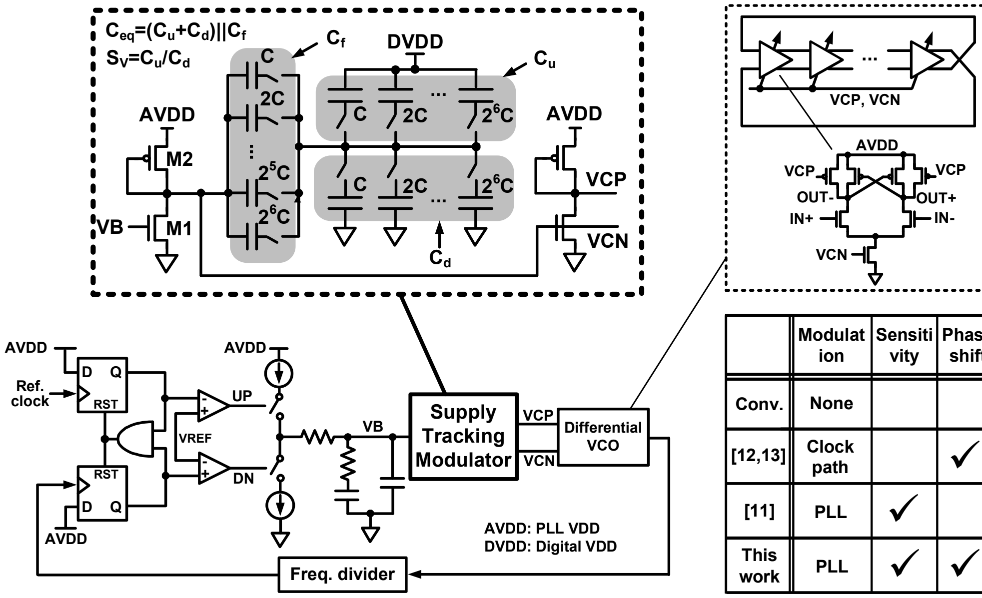
Table 4. Optimum configurations and performance of the proposed PLL for different clock distribution designs (f_{clk}=1.2GHz, T_{cp}=1ns)

Clock tree design	Supply noise frequency	Clock path property		Optim. PLL config.		Worst-case slack w/ conv. PLL	Worst-case slack w/ prop. PLL
		θ_{cp}	s_{cp}/s_d	θ_{PLL}	s_{PLL}/s_d		
#1 [21]	150 MHz	0.44π	0.2	0.30π	1	-190	-5
#2 [22]	40 MHz	0	0.7	0.05π	1.05	-204 ps	-5 ps
#3 [23]	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

6.3 Adaptive phase-shifting PLL

Fig. 37 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 it can be seen from the schematic, the capacitor banks and transistors M1 and M2 actually form a high-pass filter so that the resonant supply noise can be AC coupled to the bias voltage of the VCO to generate the adaptive clock signal.

By programming proper configurations of the three capacitor banks, the desired phase shift and noise sensitivity can be achieved.



	Modulation	Sensitivity	Phase shift	Programmable	1st droop
Conv.	None				
[12,13]	Clock path		✓		✓
[11]	PLL	✓			✓
This work	PLL	✓	✓	✓	✓

Fig. 37. Schematic of the proposed adaptive phase-shifting PLL design

A detailed analysis on how the three capacitor banks work is provided in Fig. 38. With the help of Thevenin's theorem, the impact of the capacitors banks and the resonant supply noise can be analyzed 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 output voltage when the output is open and calculating the equivalent impedance when the V_{AC} is shorted. Fig. 38(b) and 38(c) show the circuit schematics used to derive V_{eq} and Z_{eq} and the resulting expressions, respectively. As it is derived from Fig. 38(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 $S_V=C_u/C_d$, respectively, which are both digitally programmable.

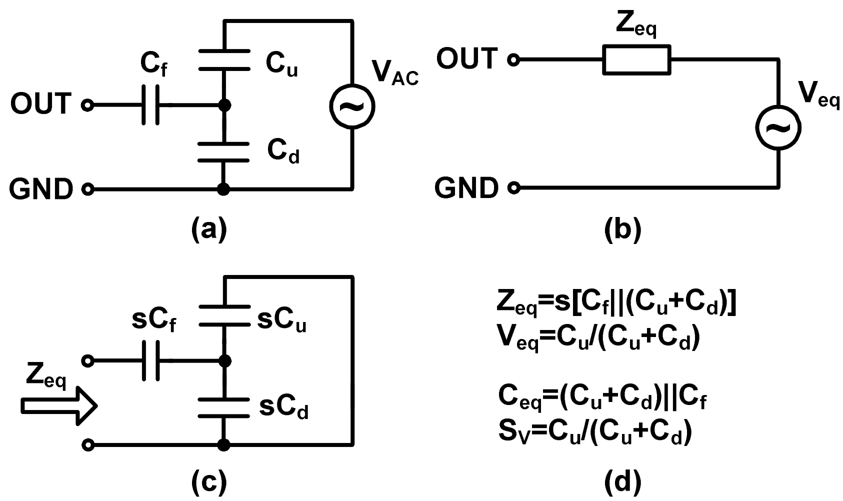


Fig. 38. Analysis of the capacitor banks with using Thevenin's theorem

Fig. 39 shows the simulation results illustrating how the supply noise sensitivity and the phase shift can be programmed. As indicated from Fig. 38(d), the supply noise sensitivity S_v can be easily programmed by selecting different ratios between C_u and C_d. Note that in order to keep the phase shift unchanged while adjusting S_v, the sum of C_u and C_d needs to be kept constant. On the other hand, it is difficult to program the phase shift without affecting the supply noise sensitivity. This is because the phase shift is introduced by a high-pass filter and can only be adjusted by changing the equivalent capacitance C_{eq}. Clearly, any change in C_{eq} will affect both the phase shift and the amplitude of the output. In this work, we always change C_u, C_d and C_f together and keep their relative ratios unchanged when programming the phase shift value. Fig. 39 shows the simulation results of the bias voltage with different configurations for the supply noise sensitivity or the phase shift.

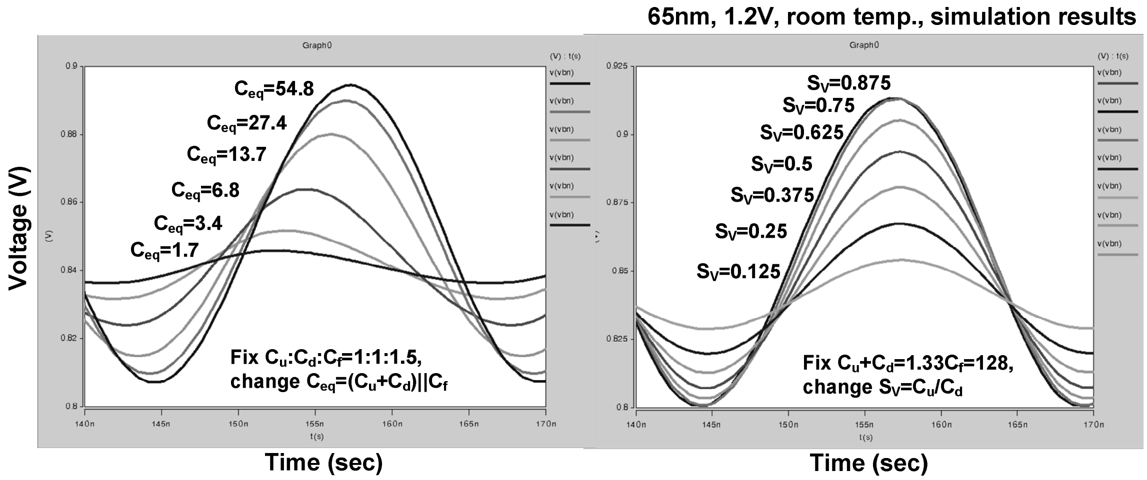
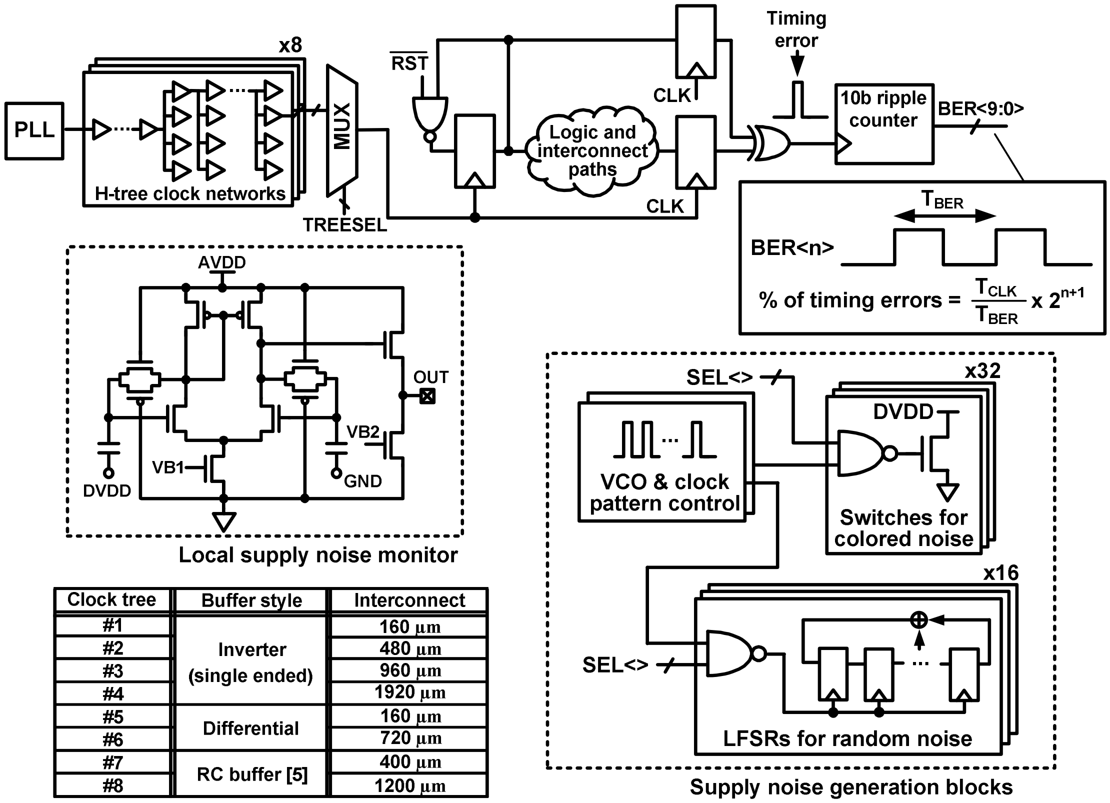


Fig. 39. Simulation results showing the programmability of the proposed PLL on supply noise sensitivity and phase shift

6.4 Test chip organization

A 1.2V, 65nm test chip was designed to verify the effectiveness of the proposed PLL (Fig. 40). The adaptive clock signal is generated by the PLL and then propagates through the clock distribution networks. We have implemented eight different clock trees using regular inverters, differential buffers or RC-filtered buffers [22][23] with different interconnect lengths. The schematic of the differential buffers and RC-filtered buffers are given in Fig. 41. A separate 40pF 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 $2i$

sampling 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. 42 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 -2.5dB in a large frequency range between 3MHz and 1GHz. The static power consumption of this sensor is 2.1mW.



Clock tree	Buffer style	Interconnect
#1	Inverter (single ended)	160 μm
#2		480 μm
#3		960 μm
#4		1920 μm
#5	Differential	160 μm
#6		720 μm
#7	RC buffer [5]	400 μm
#8		1200 μm

Fig. 40. Block diagram of the 65nm test chip.

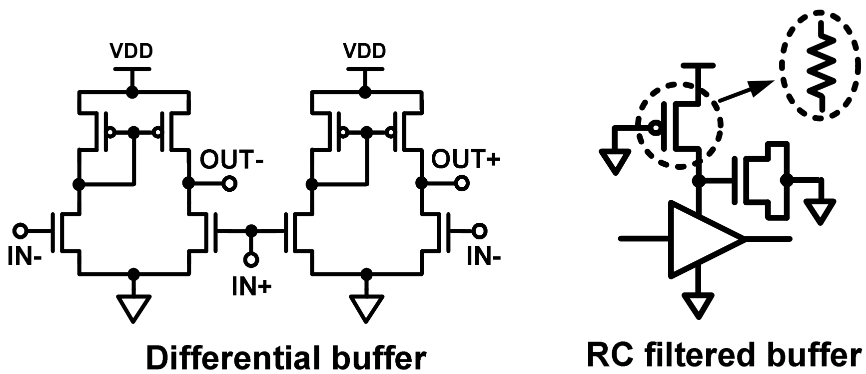


Fig. 41. Schematics of differential and RC filtered buffers.

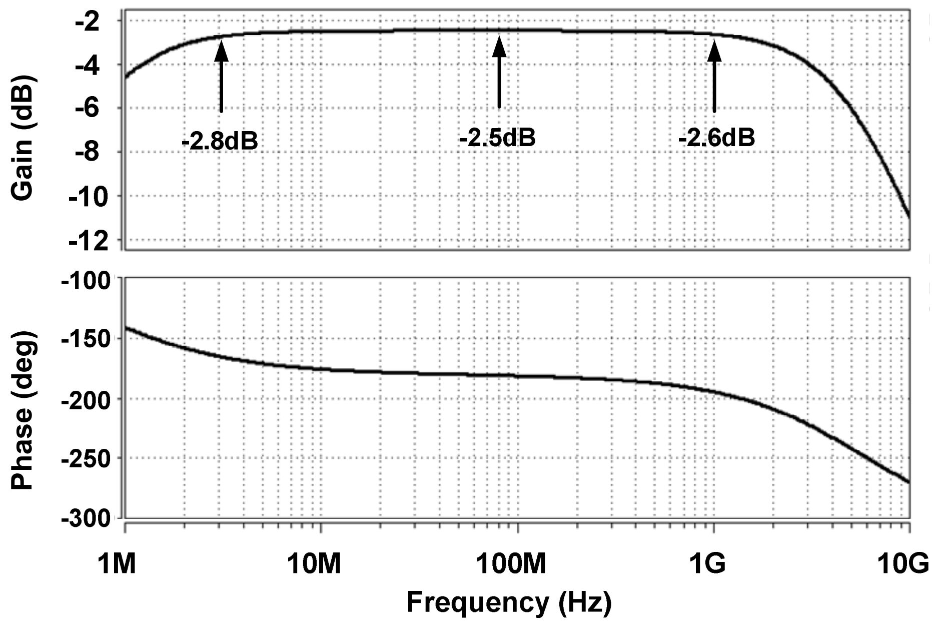


Fig. 42. Frequency response on-chip supply noise sensor.

6.5 Test chip measurement results

Figure 43(left) shows an example of the BER data measured at different clock frequencies. Without loss of generality, we define the maximum operating frequency as the point when the BER is 10^{-6}, and denote it as F_{max} in this paper. The noise waveforms measured from the supply noise monitor when injecting a first-droop noise and a sinusoidal supply noise are shown in Fig. 43(right).

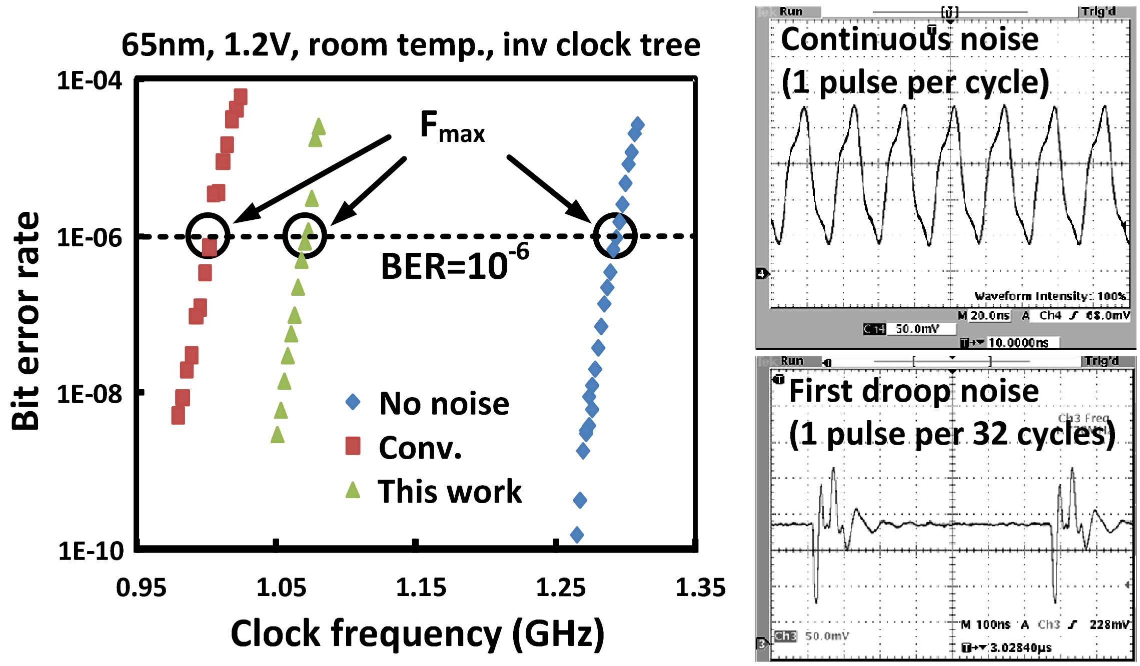


Fig. 43. Measured BER versus clock frequency (left). Example supply noise waveforms generated by noise injection circuits (right).

Fig. 44 shows the measured F_{max} while sweeping the phase shift and supply noise sensitivity values. The chip was tested for a supply voltage of 1.2V and 1.0V using a sinusoidal noise waveform. As can be seen from the figure, 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., 1.2V and 1.0V). 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.

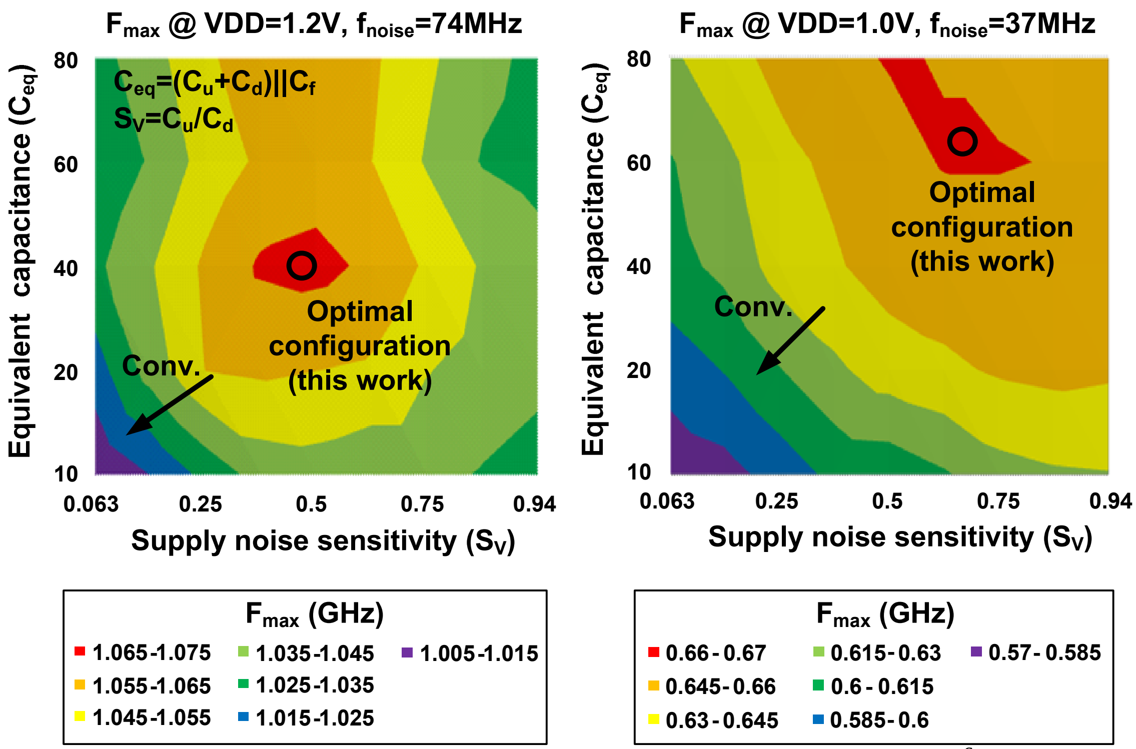


Fig. 44. Measured results at 1.2V and 1.0V showing the F_{max} (@ BER=10^{-6}) dependency on phase shift and supply noise sensitivity.

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. 45(left) show a 4% F_{max} improvement for noise frequencies between 40MHz and 300MHz. As the noise frequency increases, the performance improvement becomes smaller. This is because the clock distribution 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 1.0V supply voltage and the results also show similar performance improvement as shown in Fig. 45(right).

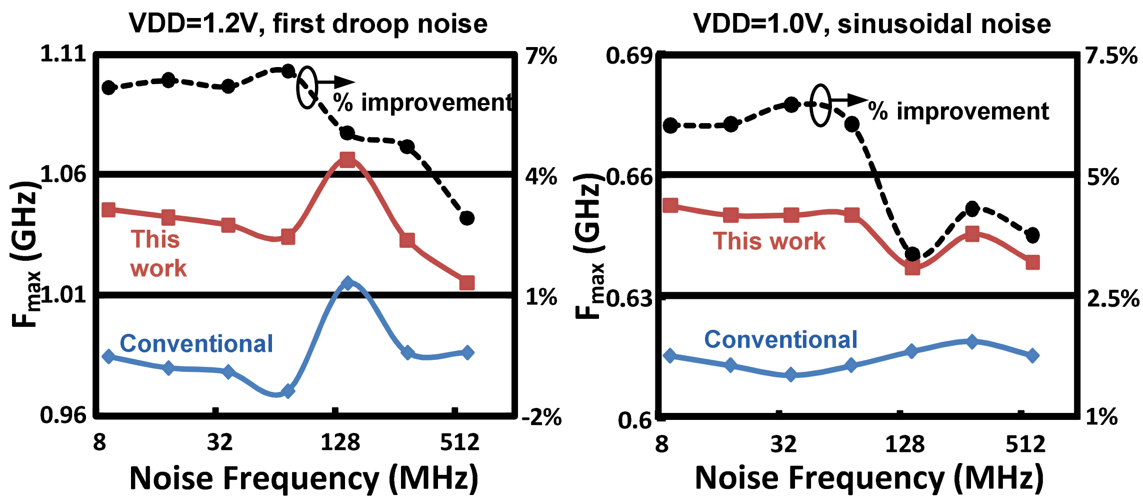


Fig. 45. Measured F_{max} at 1.2V and 1.0V for different noise frequencies.

Different clock trees were also tested and the results are shown in Fig. 46(left). Here, clock tree names with "_C" have a 40pF decap enabled in the clock tree supply and "short" or "long" refers to the interconnect length between the clock buffers. For a 74MHz 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 tested with the first-droop noise injected at 37MHz under a 1.0V supply voltage. As can be seen from measurement results shown in Fig. 46(right), a 3.3% to 6.8% improvement on F_{max} has been achieved with different clock tree designs by introducing the proposed adaptive phase-shifting PLL.

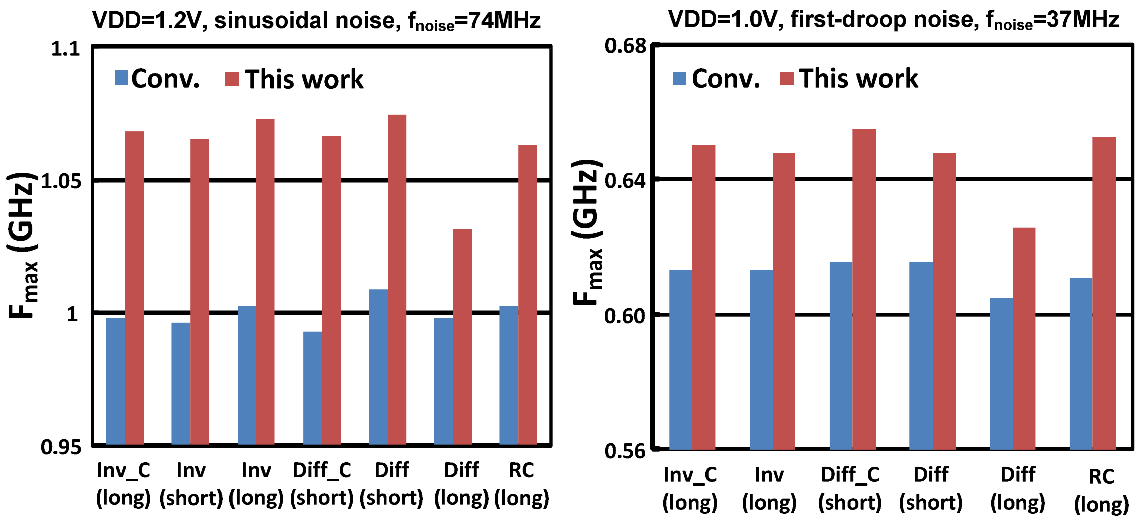
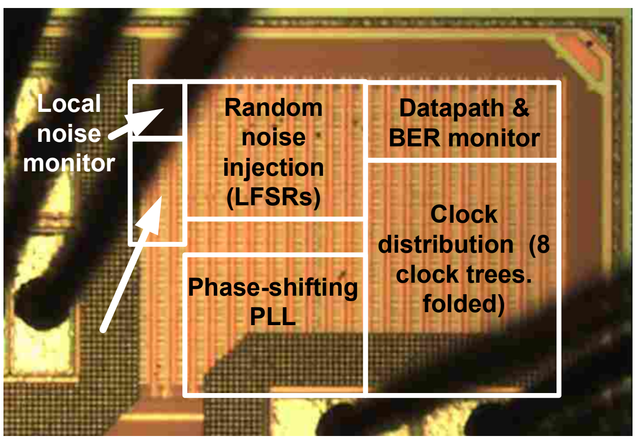
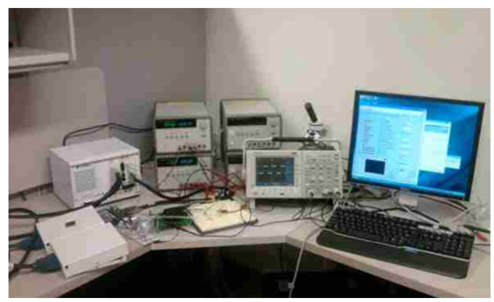


Fig. 46. Measured F_{max} at 1.2V and 1.0V for different clock trees.

The chip microphotograph and the chip performance summary are provided in Fig. 47.





Technology	65nm LP CMOS	Supply voltage	1.2V
Total area	350 x 250 μm^2	PLL area	120 x 100 μm^2
Regulation frequency	40-300MHz	F_{max} impr-ovement	3.4%-7.3%

Fig. 47. Chip micrograph and performance summary of the test chip.

6.6 Simulation results on 32nm process

To further validate the effectiveness of the proposed adaptive phase-shifting PLL, we designed such a PLL in a 32nm CMOS process and simulated its performance with several different clock distribution designs. Fig. 48 shows the schematic of the test circuit comprising a proposed phase-shifting PLL operating at 2.58GHz, a 16-stage FO4 inverter chain datapath and a 20-stage clock buffer chain with a nominal delay of 1.0ns. For easier control on the clock path characteristics, the amplitude and the timing offset of the supply noise seen by the clock path were adjusted in simulations to emulate the behaviors of the 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. 49. As shown on the top left of this figure, for the clock path with the same noise sensitivity as the datapath (s_{cp}=1.0s_d and θ_{cp}=0.0π), the best timing slack is achieved at the maximum filtering capacitance (C_{eq}). This means that no phase shift is needed in the PLL, which is consistent with the modeling results shown in Table 4. Similarly, the performance of the proposed PLL was simulated for a few other clock paths. As we can see from the figure, by optimizing the filtering capacitance (C_{eq}) and the supply noise sensitivity (S_v) of the proposed PLL, the worst-case timing slack can be improved by 27-47ps (7.1%-12.2% of clock period) for various clock trees,

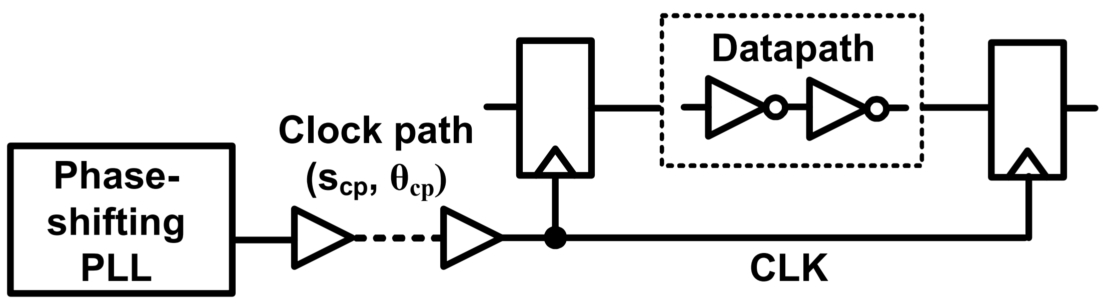


Fig. 48. Schematic of the test circuit used for validating the performance of the proposed PLL in 32nm CMOS process.

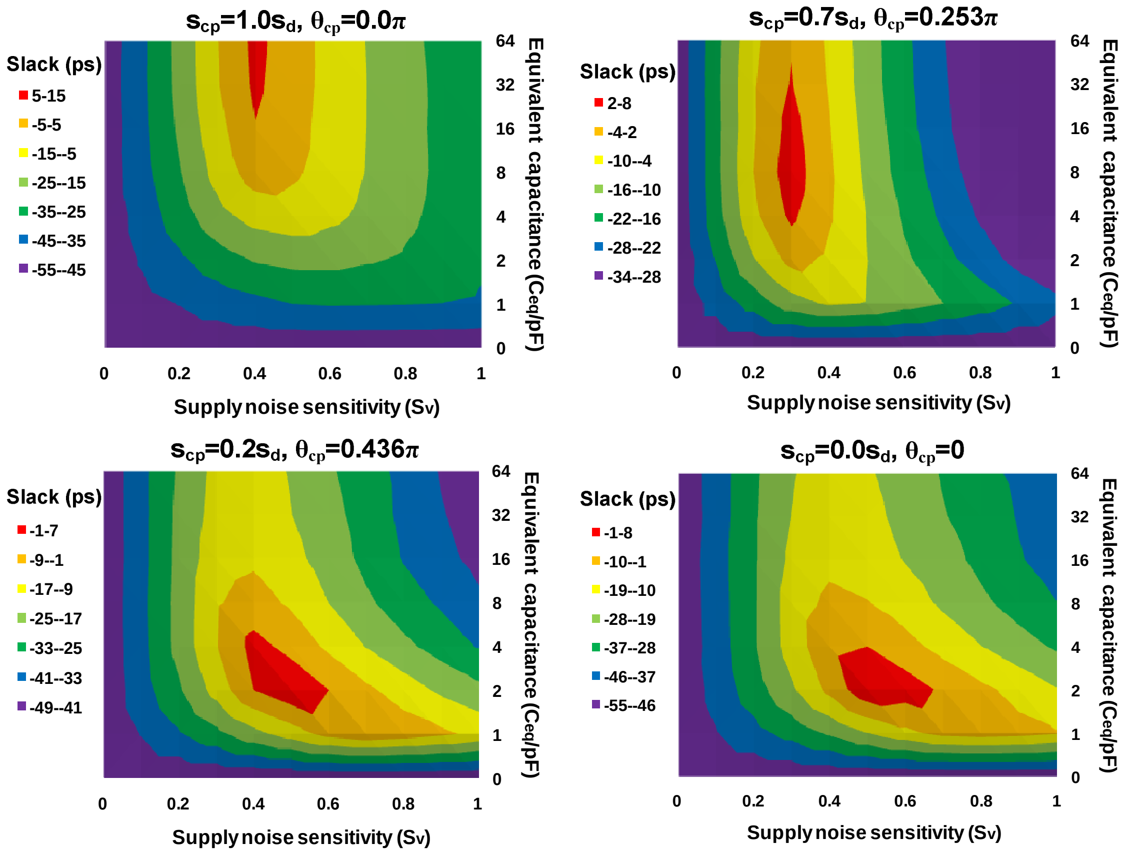


Fig. 49. Simulated timing slack with different configurations of the PLL for different clock trees.

Chapter 7

IR NOISE REDUCTION IN MULTI-CORE SYSTEMS

In this section, we will investigate another import source of the supply noise, IR noise. Then we propose to use switched capacitor DC/DC converters for IR noise reduction in multi-core systems.

7.1 IR noise and dynamic voltage and frequency scaling

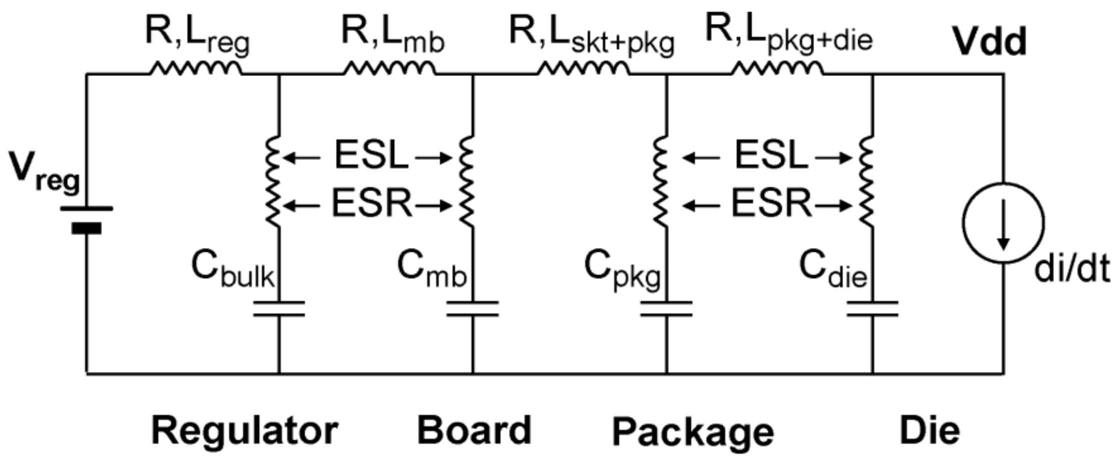


Fig. 50. A simplified model for the power delivery systems in microprocessors [22]

Fig. 50 shows a simplified model for the power delivery systems in microprocessors [22]. As it has been discussed in Chapter I, the bonding/packaging inductance and the die capacitance form a LC tank and will cause the resonant supply noise, which typically resides in the 40MHz to 300MHz frequency band. On the other hand, as shown in Fig. 50, the parasitic resistance in the power delivery system can introduce IR drop in the supply voltage, which can cause large performance degradation if the total amount of current is large.

In recent years, Dynamic Voltage and Frequency Scaling (DVFS) has become a popular approach to improve the performance of microprocessors, especially for multi-core processors, while keeping an acceptable power consumption budget [28][29][30]. When DVFS is applied in a multi-core system, each core can run at different supply voltage and operating frequency depending on its own work load. For example, if there is a high-priority task that be parallelized, several cores will operate at high supply voltages and high frequencies to get the task done quickly. In another case, if the high-priority task cannot be parallelized, the DVFS system will choose one of the cores to operate at high supply voltage and high frequency while keep other cores in idle modes.

7.2 IR noise reduction with current borrowing

As it has been explained in the previous section, a large current will lead to a large IR drop in the supply voltage and thus will degrade the performance of the microprocessor. Fig. 51 shows a simplified circuit model for the power delivery in a dual-core processor. Assume one core C1 (VDD1, C_{VDD1}, I_{VDD1}) in the multi-core system is consuming a large current (I_{VDD1}), the parasitic resistance will introduce a large IR drop on VDD1, which will degrade the performance of C1. On the other hand, despite the large current consumption from VDD1, the adjacent cores, however, might work in a light load mode, or even idle mode. Therefore, if C1 can “borrow” some current from those adjacent cores, the IR drop on VDD1 can be reduced because of the smaller current flowing through R_{VDD1}. On the other hand, the borrowed current will lead to extra IR drop on those adjacent cores providing current to C1, but the performance degradation in those cores will be small because they are running at light-load or idle modes.

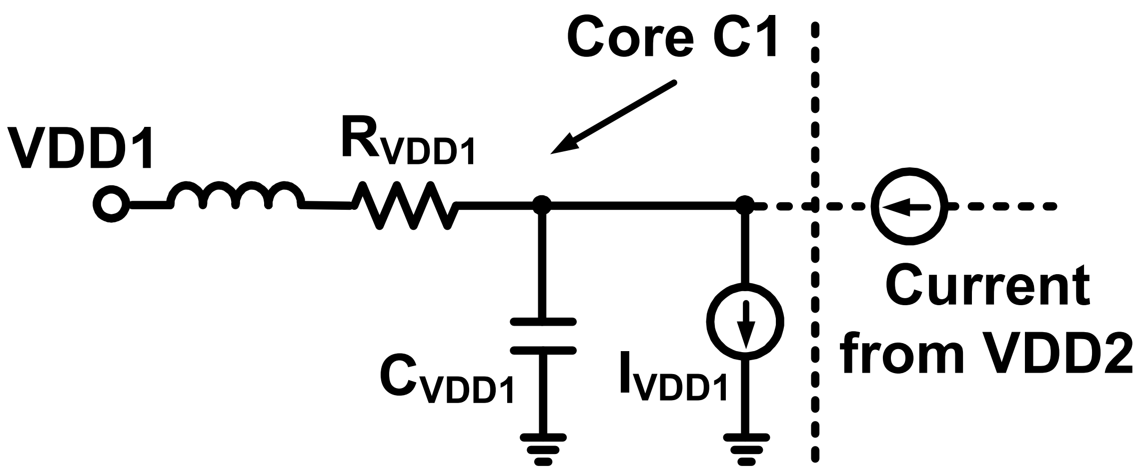


Fig. 51. IR noise reduction current borrowing.

One thing to note here is that the supply voltage of an adjacent core (e.g., VDD2 as shown in Fig. 51) can be lower than VDD1 due to the nature of DVFS. Therefore, the voltage level of VDD2 must be boosted to be higher than VDD1 before it can provide current to C1. Moreover, the current borrowing should be able to work on both directions, i.e., current should be able to flow from VDD1 to VDD2 or vice versa. Based on above observations, we propose to use bi-directional voltage doublers to achieve this goal and the schematic is shown in Fig. 52. Compared with a conventional voltage doubler, two pairs of switches are added to control the flowing direction of the current.

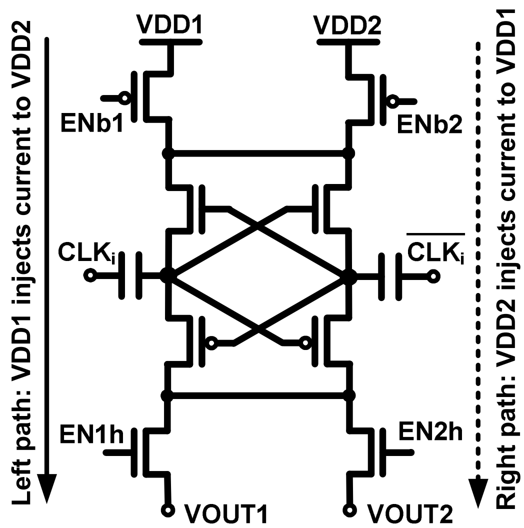


Fig. 52. Schematic of the proposed bi-directional voltage doubler.

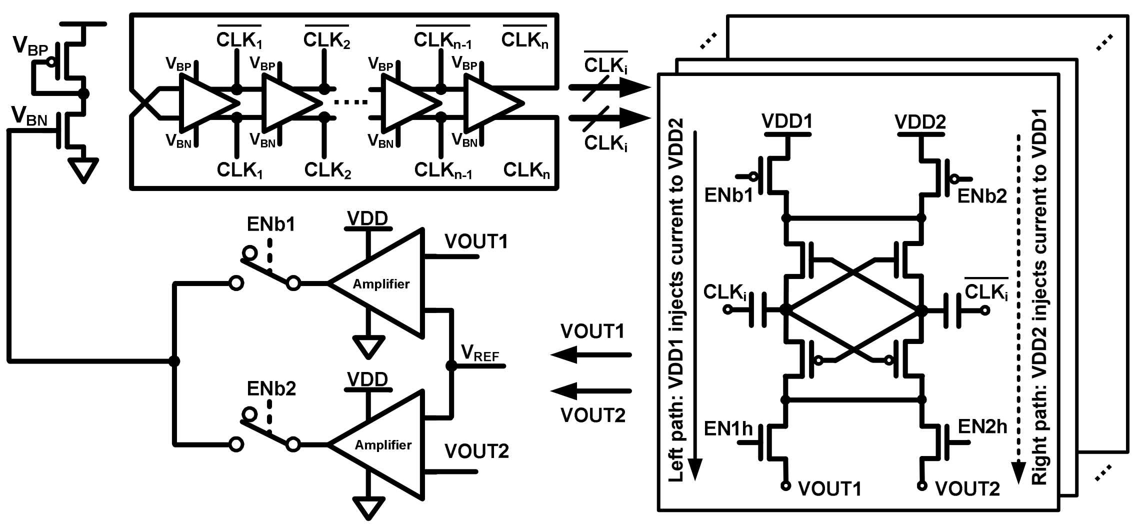


Fig. 53. Schematic of the proposed bi-directional high power-density switched capacitor DC/DC converter with closed-loop control

The proposed switched capacitor (SC) DC/DC converter consists of three major blocks: a voltage doubling block, a differential voltage-controlled oscillator (VCO) and a feedback control block. Fig. 53 shows the simplified schematic of the proposed converter design.

As shown in Fig. 53, modified Favrat cells are used for voltage doubling [31][32]. Switches are added to the cells to enable bi-directional operations. By controlling these switches, the voltage doublers can work in three different modes: (1) V_{DD1} provides current to boost V_{DD2}; (2) V_{DD2} provides current to boost V_{DD1}; (3) and a disabled mode. Note that the voltage levels of the control signals *EN1h* and *EN2h* have to be shifted to between VDD and VDD*2 to avoid high voltage stress across the two output NMOSs.

A differential VCO is introduced to generate multi-phase complementary clock signals which drive the voltage doublers. The number of stages of the VCO is selected as large as possible to achieve better multi-phase interleaving for the voltage doubling block [33][34]. On the other hand, it should also satisfy the requirement of the maximum operating frequency, which is determined by the trade-off between power density and efficiency. The power consumption of the VCO needs to be minimized to optimize the overall efficiency of the proposed converter.

The two outputs of the voltage doublers are fed into two separate differential amplifiers. Depending on the mode of the DC/DC converter, the output of one amplifier is selected to control the bias voltage of the VCO. This configuration forms closed-loop control and thus could fix the output level (V_{OUT1} or V_{OUT2}) at a desired level by dynamically adjusting the output current of the proposed converter.

7.3 Simulation results of the proposed scheme

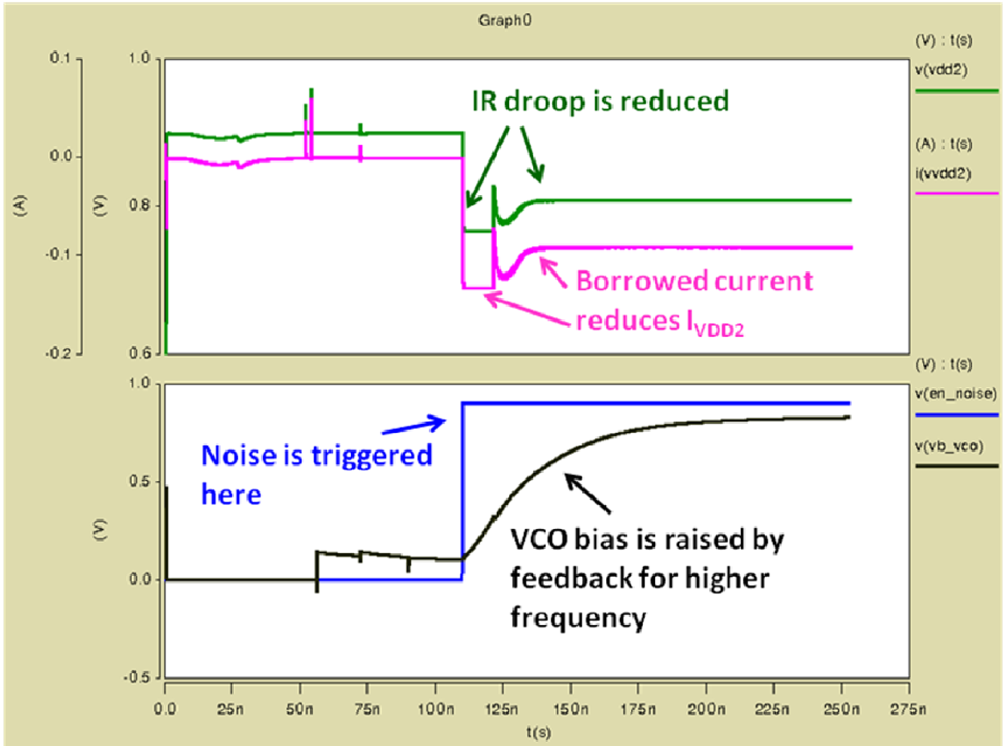


Fig. 54. Simulated performance of the proposed current borrowing scheme

The proposed current borrowing scheme with switched capacitor DC/DC converters is implemented in an industrial 32nm SOI process and the simulated performance is shown in Fig. 54. As it can be seen from the figure, one core of the process initially runs at idle mode, so the supply voltage remains constant around its nominal value 0.9V and the current is almost zero. At t=110ns, this core switches into high performance mode. A large current is drawing from the supply VDD2 and thus leads to an IR drop of 130mV. At the same time, the supply voltage sensor starts responding to the IR drop and gradually adjusts the bias voltage of the VCO to make it run at a high frequency so that the switched capacitor DC/DC converter can borrow more current from the adjacent cores. As a result, the current consumption from VDD2 is reduced from 130mA to 90mA with the help of the "borrowed" current and the IR drop is also improved from 130mV to 90mV accordingly.

Fig. 55 shows the simulation results for another more complicated case demonstrating the bi-directional operations with closed-loop control. As we can see from the waveforms, a large current I_{VDD2} occurred at t=120ns and thus caused about 150mV IR drop on VDD2. Then the supply sensor responded quickly and raised the bias voltage of the VCO to borrow more current from VDD1. Similarly at t=550ns, a large current was drained from VDD1. Again, the supply sensor raised the bias voltage of the VCO so that the IR drop can be reduced.

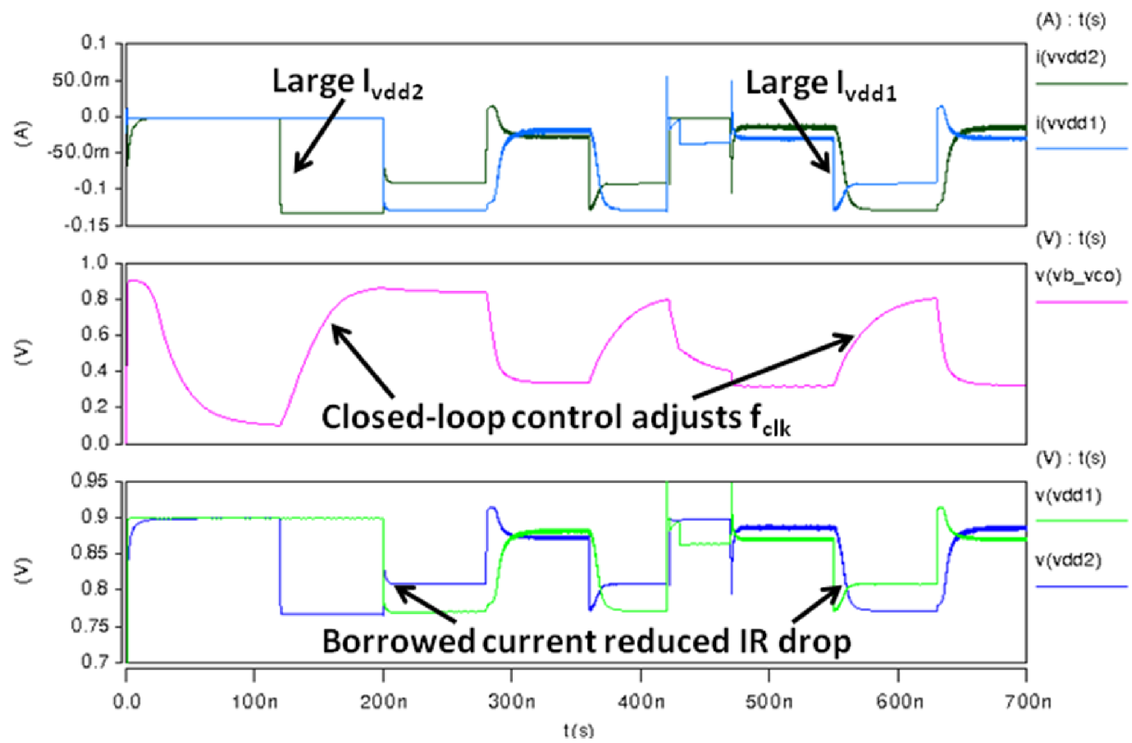


Fig. 55. Simulation results demonstrating the bi-directional operations with closed-loop control.

Chapter 8

CONCLUSIONS

In this thesis, we present a comprehensive study on the timing compensation effect between the clock cycle and the datapath delay in the presence of resonant supply noise for typical pipeline circuits. A novel phase-shifted clock distribution design and a novel adaptive phase-shifting PLL were proposed to enhance this clock data compensation effect. Compared with conventional approaches, the proposed phase-shifted clock distribution designs save 85% of the clock buffer area while achieving a similar amount of improvement in the maximum operating frequency (F_{max}) for typical pipeline circuits. In the proposed adaptive phase-shifting PLL design, both the supply noise sensitivity and the phase shift of the PLL output can be digitally programmed such that the optimal timing compensation can be achieve under different operating conditions. A mathematical framework for simulating the performance of the proposed PLL for different clock distribution designs is also presented. Two 1.2V, 65nm test chips demonstrated that the proposed phase-shifted clock distribution designs can provide an 8-27% performance improvement in F_{max} for typical resonant noise frequencies from 100MHz to 300MHz and the proposed phase-shifting PLL can provide 3-7% improvement in F_{max} under various operating conditions.

REFERENCE

[1] M. Saint-Laurent and M. Swaminathan, "Impact of Power-Supply Noise on Timing in High-Frequency Microprocessors," *IEEE Transactions on Advanced Packaging*, vol. 27, no. 1, pp. 135-144, 2004.

[2] J. M. Rabaey, A. Chandrakasan and B. Nikolic, Digital Integrated Circuits A Design Perspective, 2003.

[3] M. D. Pant, P. Pant and D. S. Wills, "On-Chip Decoupling Capacitor Optimization Using Architectural Level Prediction," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 10, no. 3, pp. 319-326, 2002.

[4] J. Xu, P. Hazucha, M. Huang, et al., "On-Die Supply-Resonance Suppression Using Band-Limited Active Damping," *International Solid-State Circuits Conference (ISSCC) Dig. Tech. Papers*, pp.2238-2245, 2007.

[5] J. Gu, R. Harjani and C. Kim, "Distributed Active Decoupling Capacitors for On-Chip Supply Noise Cancellation in Digital VLSI Circuits," *Symposium on VLSI Circuits*, pp. 216-217, 2006.

[6] M. Mansuri and C. K. Yang, "A Low-Power Adaptive Bandwidth PLL and Clock Buffer With Supply-Noise Compensation," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1804-1812, 2003.

[7] L. H. Chen, M. Marek-Sadowska and F. Brewer, "Coping with Buffer Delay Change Due to Power and Ground Noise," *Design Automation Conference*, pp. 860-865, 2002.

[8] T. Fischer, J. Desai, B. Doyle, et al., "A 90-nm Variable Frequency Clock System for a Power-Managed Itanium Architecture Processor," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp.218-228, 2006.

[9] S. Yasuda and S. Fujita, “Compact Fault Recovering Flip-Flop with Adjusting Clock Timing Triggered by Error Detection,” *IEEE Custom Integrated Circuits Conference*, pp.721-724, 2007.

[10] N. Agarwal and S. S. Rath, "Low-jitter clock distribution circuit," US Patent 6,842,136 B1, Jan. 11, 2005.

[11] M. Saint-Laurent, “Clock distribution network using feedback for skew compensation and jitter filtering,” US Patent 7,317,342 B2, Jan. 8, 2008.

[12] V. Gutnik and A. Chandrakasan, "Clock distribution circuits and methods of operating same that use multiple clock circuits connected by phase detector circuits to generate and synchronize local clock signals," US Patent 7,571,359 B2, Aug. 4, 2009.

[13] J. Gu, H. Eom and C.H. Kim, "On-chip Supply Noise Regulation Using a Low Power Digital Switched Decoupling Capacitor Circuit," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 6, pp. 1765-1775, Jun. 2009.

[14] 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*, Feb. 2006, pp. 2238-2245.

[15] M. Mansuri and C.K. Yang, "A Low-Power adaptive bandwidth PLL and clock buffer with supply-Noise Compensation," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 1804-1812, Nov. 2003.

[16] S.C. Chan, P.J. Restle, T.J. Bucelot, et al, "A Resonant Global Clock Distribution for the Cell Broadband Engine Processor," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 64-72, Jan. 2009.

[17] X. Zheng and K.L. Shepard, "Design and Analysis of Actively-Deskewed Resonant Clock Networks," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 558-568, Feb. 2009.

[18] T. Ebuchi, Y. Komatsu, T. Okamoto, et al, "A 125-1250 MHz Process-Independent Adaptive Bandwidth Spread Spectrum Clock Generator With Digital Controlled Self-Calibration," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 3, pp. 763-774, Mar. 2009.

[19] D. Chan and M.R. Guthaus, "Analysis of Power Supply Induced Jitter in Actively De-skewed Multi-Core Systems", in *Int. Symp. on Quality Electronic Design (ISQED),* pp. 785-790, Mar. 2010

[20] D. Wendel, R. Kalla, R. Cargoni, et al., “The Implementation of POWER7™: A Highly Parallel and Scalable Multi-Core High-End Server Processor,” in *IEEE Int. Solid State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 102-103, Feb. 2010.

[21] N. Kurd, P. Mosalikanti, M. Neidengard, J. Douglas and R. Kumar, "Next generation Intel® core™ micro-architecture (Nehalem) clocking," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 4, pp. 1121-1129, Apr. 2009.

[22] K. L. Wong, T. Rahal-Arabi, M. Ma and G. Taylor, "Enhancing microprocessor immunity to power supply noise with clock-data compensation," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 4, pp. 749-758, Apr. 2006.

[23] D. Jiao, J. Gu, P. Jain and C. Kim, "Enhancing beneficial jitter using phase-shifted clock distribution," in *Proc. IEEE Int. Symp. Low Power Electronics and Design (ISLPED),* Aug. 2008, pp. 21-26.

[24] D. Jiao, J. Gu and C. H. Kim, "Circuit Design and Modeling Techniques for Enhancing the Clock-Data Compensation Effect under Resonant Supply Noise," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 10, pp. 2130-2141, Oct. 2010.

[25] 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 Journal of Solid-State Circuits,* vol. 36, no. 11, pp. 1647-1653, Nov. 2001.

[26] J. Jang, O. Franza and W. Burleson, "Compact Expressions for Supply Noise Induced Period Jitter of Global Binary Clock Trees," *IEEE T. on Very Large Scale Integration (VLSI) Systems*, Dec. 2010

[27] J. M. Hart, K. T. Lee, D. Chen, et al, "Implementation of a fourth-generation 1.8-GHz dual-core SPARC V9 microprocessor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 210-217, Jan. 2006.

[28] A. Allen, J. Desai, F. Verdico, et al, "Dynamic Frequency-Switching Clock System on A Quad-Core Itanium Processor", in *IEEE Int. Solid-State Circuits Conf. (ISSCC), Dig. Tech. Papers,* Feb. 2009.

[29] S. Dighe, S.R. Vangal, P. Aseron, et al, "Within-Die Variation-Aware Dynamic-Voltage-Frequency-Scaling With Optimal Core Allocation and Thread Hopping for the 80-Core TeraFLOPS Processor", *IEEE J. Solid-State Circuits,* vol. 46, no. 1, pp. 184-193, Jan. 2011.

[30] K.J. Nowka, G.D. Carpenter, E.W. MacDonald, et al, "A 32-bit PowerPC System-On-A-Chip with Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling", *IEEE J. Solid-State Circuits,* vol. 37, no. 11, pp. 1441-1447, Nov. 2002.

[31] P. Favrat, P. Deval, and M. Declercq, “A High-Efficiency CMOS Voltage Doubler”, *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 410-416, Mar. 1998.

[32] K. Phang and D. Johns, “A 1V 1mW CMOS front-end with on-chip dynamic gate biasing for a 75Mb/s optical receiver”, *in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 218-219, Feb. 2001.

[33] D. Somasekhar, B. Srinivasan, G. Pandya, et al, “Multi-Phase 1 GHz Voltage Doubler Charge Pump in 32 nm Logic Process”, *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 751-758, Apr. 2010.

[34] T.V. Breussegem and M.Steyaert, “A 82% Efficiency 0.5% Ripple 16-Phase Fully Integrated Capacitive Voltage Doubler”, *in Symposium on VLSI Circuits*, pp. 198-199, Aug. 2009.