Stochastic Current Prediction Enabled Frequency Actuator for Runtime Resonance Noise Reduction [*]

Yiyu Shi[†] Jinjun Xiong[*]
[†]Electrical Engineering Dept., UCLA
[†] Los Angeles, California, 90024
[†]{yshi, lhe}@ee.ucla.edu

Howard Chen[*] Lei He[†]
[*]IBM Thomas J. Watson Research Center
[*]Yorktown Heights, New York, 10598
[*]{jinjun,haowei}@us.ibm.com

ABSTRACT

Power delivery network (PDN) is a distributed RLC network with its dominant resonance frequency in the low-to-middle frequency range. Though high-performance chips' working frequencies are much higher than this resonance frequency in general, chip runtime loading frequency is not. When a chip executes a chunk of instructions repeatedly, the induced current load may have harmonic components close to this resonance frequency, causing excessive power integrity degradation. Existing PDN design solutions are, however, mainly targeted at reducing high-frequency noise and not effective to suppress such resonance noise. In this work, we propose a novel approach to proactively suppress this type of noise. A method based on a high dimension generalized Markov process is developed to predict current load variation. Based on such prediction, a clock frequency actuator design is proposed to proactively select an optimal clock frequency to suppress the resonance. To the best of our knowledge, this is the first in-depth study on proactively reducing runtime instruction execution induced PDN resonance noise.

1. INTRODUCTION

Two dominant types of noise are present in a power delivery network (PDN): *peak noise* and *resonance noise* [1]. Peak noise usually occurs when the instantaneous switching current load becomes maximum [2] for a short duration with its energy spectrum lying in the high-frequency range [1]. Abundant research has been done to minimize peak noise for PDN design (e.g., [3–6]).

Resonance noise is a result of the distributed RLC characteristics of a PDN, which includes parasitic inductance of interconnect and decoupling capacitance. The PDN forms a resonant tank that produces impedance peaks at multiple resonant frequencies. The dominant resonance frequency (f_{res}) usually occurs at low-to-middle frequency range (MHz to 100MHz) [7, 8]. Though high-performance chips' working frequencies are much higher than this resonance frequency in general, chip runtime loading frequency is not. When current loads exhibit a periodical rate close to f_{res} caused by a looping sequence of instruction execution, the impedance would increase significantly at this resonance frequency, causing persistent undershoots and overshoots that exceed the droop tolerance of the PDN. Resonance noise compromises chip performance, hold-time margins, and gate oxide integrity [7, 9]. Despite the importance of resonance noise for reliable PDN design, resonance noise suppression has not gained enough attention in the EDA community.

Traditional static solutions, such as adding more passive capacitors or more supply pins, are not effective to suppress the resonance noise. Hence, dynamic run-time solutions are being studied recently in the literature. For example, the authors of [9] proposed to dynamically switch on-chip decoupling capacitors to suppress resonance noise for microprocessors' PDN designs, while the authors of [7] provided an on-die resonance-suppression circuit that uses band-limited active damping to reduce resonance noise. But all these approaches are retroactive, i.e., they only remedy the noise problem when the noise problem has occurred, which is often too late as wrong values might have been latched already. Hence a better approach should proactively suppress the resonance noise when such issues are predicted to happen soon.

The major contribution of this paper is as follows. We model chip dynamic current loads as a high dimension generalized Markov process, and develop a novel stochastic method to predict the future current load based on the knowledge of existing current profile. A proactive PDN design approach is proposed to suppress resonance noise by leveraging a frequency actuator consisting of on-chip programmable PLLs and dynamic power supply current sensors [10, 11]. We develop an efficient controlling algorithm to judiciously select the run-time clock frequency so that the resonance noise is contained below the tolerance bound with minimum impact on chip performance. Compared with baseline design without frequency actuator, experimental results show that our frequency actuator design alone reduces maximum noise by 16% and average noise by 30%, while our proactive frequency actuator with current prediction reduces maximum noise by 77% and average noise by 85%. In terms of system level performance, compared with the baseline model, our frequency actuator alone can reduce the system latency overhead by up to 35% , and with current prediction it can reduce the system latency overhead by up to 93%.

The remainder of the paper is organized as follows. We motivate the study of this work in section 2, and present the problem formulation and overall design methodology in section 3. We develop the stochastic current prediction algorithm in section 4, and propose the optimum frequency selection in section 5. The experimental results are presented in section 6 and concluding remarks are given in section 7.

2. MOTIVATION

Most existing work on PDN designs models the load of a port as a *single* current spike $I_0(t)$ with a short duration time τ as shown in Fig. 1, which is typically modeled as a triangular waveform within $[0, \tau]$. Hence the *peak noise* resulting from $I_0(t)$ is essentially a high-frequency noise with its frequency in the range of $1/\tau$. For 65 nm designs, the duration τ is on the order of $100\,ps$, which produces peak noise at the frequency range on the order of 10 GHz.

[*]This paper is supported in part by an NSF CAREER award CCR0306682 and a UC MICRO grant sponsored by Actel and Fujitsu. Address comments to lhe@ee.ucla.edu.

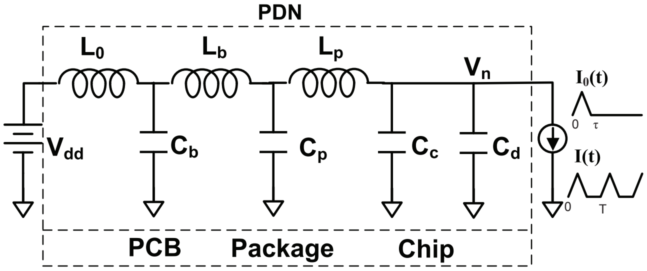

Figure 1: A simple PDN example.

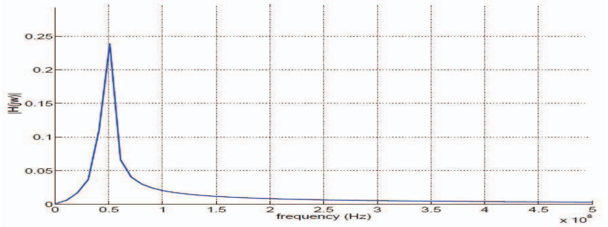

Figure 2: Impedance of the given PDN example.

But in reality, the current load $I(t)$ possesses a much lower frequency component because of the periodic nature of functional execution. Without loss of generality, we assume $I(t)$ has $I_0(t)$ as its only current component with a period of $T >> \tau$ as shown in Fig. 1; while the real case can be treated as a superposition of this simple scenario with different combinations of $I_0(t)$ and T. By performing AC analysis on the circuit, we obtain the voltage response at the port of interest as

$$V_n(j\omega) = H(j\omega)I(j\omega) = \frac{\sqrt{2\pi}}{T}H(j\omega)\sum_{k=-\infty}^{\infty} I_0(jk\omega_0)\delta(\omega - k\omega_0),$$

where $H(j\omega)$ and $I(j\omega)$ are the impedance and current load of the PDN at the port of interest, respectively, with $\omega_0 = 2\pi/T$.

Though the PDN model used in our work is a meshed RLC network, we illustrate the concept resonance noise through a simple circuit model of the PDN as shown in Fig. 1. The dominant resonance frequency f_{res} of this system is approximately given by

$$f_{res} = \frac{\omega_{res}}{2\pi} = \frac{1}{2\pi\sqrt{L_p(C_c + C_d)}}. \qquad (1)$$

where C_c is the lumped on-chip intrinsic capacitance, C_d is the decoupling capacitance, and L_p is the lumped package inductance. For some typical extracted values of an industrial design, Fig. 2 illustrates its impedance frequency response with the dominant resonance frequency around 50 MHz.

As the resonance frequency f_{res} for a typical PDN design is on the order of 100 MHz (equivalent to 20 cycles for a 2GHz CPU), it is far from the frequency range of peak noise, but rather closer to the periodic function execution frequency. When the low frequency components of $I(t)$ are close to ω_{res}, i.e.,

$$k\omega_0 = k2\pi/T \approx \omega_{res}, \qquad (2)$$

the voltage fluctuation $V_n(jk\omega_0)$ would increase significantly to cause chip malfunction. The voltage drop measured at this resonance frequency is called *resonance noise*. Since the impedance $H(j\omega)$ at the port of interest is constant for a given PDN topology, the magnitude of $V_n(j\omega)$ is propotional to the magnitude of current load $I(j\omega)$. Therefore, we also call the current load measured at this resonance frequency as resonance noise whenever there is no ambuguity.

It is generally believed that inserting decoupling capacitance can minimize noise of a PDN. This is valid only for high-frequency peak noise reduction, but not for the suppression of resonance noise. Resonance noise greatly depends on the run-time operation, as it affects the low-frequency components of $I(t)$. An effective way to minimize the low-frequency resonance noise would be to change clock period T directly so that the relation (2) does not hold.

3. PROBLEM FORMULATION

In order to control the low frequency component of the current load to avoid the resonance frequency, we need to dynamically adjust work load period T. There are two possible ways to apply the adjustment: the first one is to adjust the PLL to change clock frequency; the other one is to adjust the power supply voltage level [12]. Either approach can effectively change the duration of work load, thus achieving different frequency response. As an illustration, we choose to adjust clock frequency directly by employing a programmable PLL design similar to [13, 14] in this paper.

We assume the PLL allows a range of clock frequencies from f_{min} to f_{max}, where the chip is signed-off for f_{max}. In other words, the chip will work for any frequency below f_{max}. The actual chip performance will vary for different frequencies depending on the application. However, note that frequency adjustments are temporary, so the impact on performance is also limited, as will be verified by our experiments.

We employ on-chip current sensors to monitor the dynamic current load $I(t)$ for each clock domain of interest in the design. Based on the history $I(t)$ data, a *control unit* determines an optimal clock frequency to be generated by the programmable PLL. This procedure continues such that the low-frequency components of the current load $I(t)$ will not be close to the resonance frequency f_{res}. In other words, we keep the resonance noise below a user-specified tolerant bound. Because it takes time for the PLL to track the adjustment, it is important to select an optimal clock frequency so that the impact on performance degredation is minimum.

The control unit can be implemented as a *frequency actuator* in hardware, and it consists of two major parts. The first is a current load prediction module or *predictor*, which predicts the incoming current load profile and its impact on frequency response based on the history data of current loads. The second module is an *optimizer*, which determines an optimal clock frequency to be generated by the programmable PLL.

To reflect reality, the following design constraints are considered: (1) a finite m discrete number of clock frequencies for the programmable PLL to choose from; (2) non-instantaneous PLL tracking time, i.e., it takes certain number of clock cycles for the programmable PLL to transit from an existing clock frequency to the next one; (3) transition overhead, i.e., the PLL has to stay within each chosen clock frequency for at least a certain number of clock cycles before it can transit to the next frequency to reduce the overhead cost for frequency switching.

In the interest of space, we focus on the CAD aspects of the proposed methodology, i.e., how we predict the current profile based upon the historical current sensor data, and how this profile can be utilized to select the optimal frequency to suppress resonance noise for a PDN design. Detailed discussion on how to design power supply dynamic current sensors with minimum area and power overhead is beyond the scope of this paper. Interested readers please refer to [10, 11].

4. STOCHASTIC CURRENT PREDICTION

4.1 Current Prediction Modeling

To select an optimal frequency, we need to know how the frequency response would change for the incoming work load variations. To do so, we first need to predict how the current loads would change for the next few clock cycles.

For a given clock domain of interest, there are n number of current sensors monitoring the current load. We represent the current waveform within one clock cycle as a triangular waveform, and each current sensor records either a peak or average current value for this waveform. Such a monitored value for current sensor j at cycle k is denoted as i_k^j; in other words, there is a correspondence between a monitored current value and the triangular current waveform that it represents. We record all the currents for the same cycle as a vector I_k, i.e.,

$$I_k = [i_k^1, i_k^2, \ldots, i_k^n]^T, \tag{3}$$

where n is the total number of sensors.

Under different input vectors and working conditions, I_k would be different for different cycles. Moreover, for I_k that are close in cycles, they are highly correlated; while for I_k that are far apart in cycles, they are less (or even not) correlated. The correlation distance $\overline{D}$ is the number of cycles such that all I_k are uncorrelated when they are at least $\overline{D}$ cycles apart.

Based on these observations, we model I_k as a *generalized Markov stochastic process* over different clock cycles. A generalized Markov process is a stochastic process whose value at time k depends not only on its value at time $k-1$, but also on its values at time $k-2$, . . ., $k-Q$. These past states collectively are called the history of length Q of the process [15].

We propose to use a linear filter as the predictor to predict the current load I_k as

$$\hat{I}_k = \sum_{i=1}^{Q} \Psi_i I_{k-i}, \tag{4}$$

where Ψ_i are $n \times n$ coefficient matrices to be determined, while I_{k-i} are historical current vectors. Apparently the choice of Q, hence Q number of Ψ_i, helps to balance the trade-off between our model prediction accuracy and computation efficiency.

In this work, we set Q same as the correlation distance $\overline{D}$, as any current vectors that are $\overline{D}$ cycles apart have no correlation. Moreover, instead of using all Q historical current vectors, we sample M number of them for prediction, i.e.,

$$\hat{I}_k = \sum_{i=1}^{M} \Psi_i I_{k-i\cdot L}, \tag{5}$$

where L is the sampling separation such that $M \cdot L = Q$. In another words, we reduce the number of unknown coefficient matrices Ψ_i from Q to M. Our goal is to determine the set of M coefficient matrices Ψ_i such that (5) is a good predictor for I_k for **any** randomly selected current vectors in Q consecutive clock cycles. Mathematically, this problem can be stated as follows.

FORMULATION 1. *Given any randomly selected current vectors in $M \cdot L$ consecutive clock cycles with sampling spacing as L: I_{k-L}, $I_{k-2\cdot L}$,. . ., $I_{k-M\cdot L}$, find a set of matrices Ψ_1, Ψ_2,. . .,Ψ_M such that the expectation of the prediction error for I_k is minimized, i.e.*

$$\min_{\Psi_1,\ldots,\Psi_M} E||I_k - \hat{I}_k||_2^2 \tag{6}$$

We propose to solve (6) in two approaches with each providing different trade-offs between prediction accuracy and computation complexity (hardware area cost). Obviously, the prediction accuracy depends on both M and L, and we will report this in the experimental section.

4.2 LMS Adaptive Filter

The first approach is based on the framework of a least-mean-square (LMS) adaptive filter as illustrated in Fig. 3, where matrices Ψ_i are time-varying matrices and are dynamically adjusted at every clock cycle during runtime. We denote $\Psi_{i,k}$ as the value of Ψ_i at clock cycle k, and $\delta\Psi_{i,k}$ as the adjustment for $\Psi_{i,k}$. Then we have

$$\begin{aligned} \hat{I}_k &= \sum_{i=1}^{M} \Psi_{i,k-1} I_{k-i\cdot L}, & (7)\\ \delta\Psi_{i,k} &= \mu \cdot I_{k-i\cdot L} \cdot e_k^T, & (8)\\ \Psi_{i,k} &= \Psi_{i,k-1} + \delta\Psi_{i,k}, & (9) \end{aligned}$$

where μ is the step size determined by experiments and $e_k = I_k - \hat{I}_k$ is the prediction error.

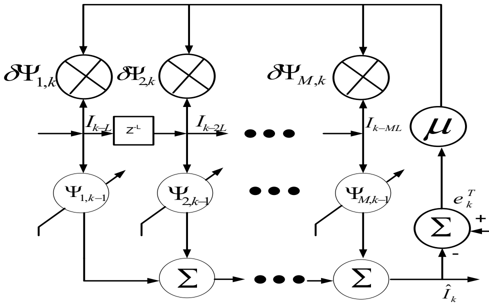

Figure 3: Framework of an LMS adaptive filter.

In general, the LMS adaptive filter approach is accurate as a predictor, because it can automatically adjust itself to follow large changes in statistical behavior of the sequence of current vectors, and is thus suitable for systems with diverse operations. But hardware implementation cost of this type is relatively high; and it cannot always guarantee the convergence of the coefficients Ψ_i^k in all situations [16]. Therefore, we propose a second approach called *predetermined linear filter* as an alternative solution.

4.3 Predetermined Linear Filter

The idea of a predetermined linear filter is based upon the off-line simulation and uses the simulation results as training data to find an optimal set of constant matrices Ψ_i to the problem of (6). Similar to the vectorless P/G analysis in [17], we assume that each clock domain under study is partitioned into blocks such that different blocks are relatively independent. For each block, there are multiple ports connected to the power network, and each port is modeled as a time-varying current load for the power network. We apply extensive simulation to each block *independently* to get the current signatures for all ports, which are then aggregated to obtain the current signature for the n current senors for this clock domain. After extensive simulation, we have current vectors at many different clock cycles. We then need to determine a set of Ψ_i to (6) based on these simulation data. To do so, we present the following theorem with the detailed proof omitted because of limited space.

THEOREM 1. *If we define a set of matrices $\mathbf{r}_{i,j} = E(I_i I_j^T)$,* *then the optimal Ψ_i $(1 \le i \le M)$ of (6) are*

$$\Psi_i = \mathbf{S} \times \mathbf{R^{-1}} \times \mathbf{e_i}, \tag{10}$$

where $S \in R^{n \times M \cdot L}$, $R \in R^{M \cdot L \times M \cdot L}$, and $e_i \in R^{M \cdot L \times n}$, and are given by

$$\mathbf{S} = \left[\begin{array}{cccc} r_{k,k-L} & r_{k,k-2L} & \ldots & r_{k,k-M \cdot L} \end{array} \right], \tag{11}$$

$$\mathbf{R} = \left[\begin{array}{cccc} \mathbf{r}_{k-L,k-L} & \mathbf{r}_{k-L,k-2L} & \cdots & \mathbf{r}_{k-L,k-M \cdot L} \\ \mathbf{r}_{k-2L,k-L} & \mathbf{r}_{k-2L,k-L} & \cdots & \mathbf{r}_{k-M \cdot L,k-L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_{k-M \cdot L,k-L} & \mathbf{r}_{k-M \cdot L,k-2L} & \cdots & \mathbf{r}_{k-M \cdot L,k-M \cdot L} \end{array} \right], \tag{12}$$

$$\mathbf{e}_i = \left[\begin{array}{cccccc} \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I} & \mathbf{0} & \ldots \end{array} \right]^T, \tag{13}$$

with $\mathbf{I}$ being an $n \times n$ identity matrix at the i^{th} block matrix of e_i.

Without going into too much details, we note that $\mathbf{R}$ is closely related to two types of correlation for current vectors. Specifically, (1) *logic-induced correlation*, i.e., current loads at different location are correlated and cannot reach the maximum at the same time due to the inherent logic dependency for a given design; and (2) *temporal correlation*, i.e., for current loads at the same port, they cannot attain the maximum value at all time, and depending on the functionality being performed, the current variations for different clock cycles are correlated. The element at m^{th} row and n^{th} column of the block matrix $\mathbf{r}_{i,j}$ actually reflects the logic-induced correlation between location m and n and the temporal correlation between clock cycle i and j. Therefore, the matrix $\mathbf{R}$ actually characterizes both the logic-induced correlation over all n locations and the temporal correlation over all clock cycles.

Once we obtain Ψ_i according to (10), at any clock cycle k, we can predict future L cycles' current vectors $\hat{I}_k$, $\hat{I}_{k+1}$, ..., $\hat{I}_{k+L-1}$ by using the $M \cdot L$ history current loads I_{k-1},I_{k-2}, ...,$I_{k-M \cdot L}$ as

$$\hat{I}_{k+l} = \sum_{i=1}^{M} \Psi_i I_{k+l-i \cdot L}, \quad 0 \le l \le L-1 \tag{14}$$

As there exists correspondence between a triangular waveform model and a recorded (or predicted) current value, we can reconstruct future K cycles' current waveform $u(t)$ for all $K \le L$ as

$$\begin{aligned} u(t) &= \sum_{i=1}^{K} \hat{I}_{k+i} u_\triangle(t-(i-1) \times T) \\ &= \sum_{i=1}^{K} u_i(t-(i-1) \times T), \end{aligned} \tag{15}$$

where T is the clock period, $u_\triangle(t)$ is a triangular waveform whose starting time is at the beginning of each clock cycle k with a unit peak current value; and $u_i(t) = \hat{I}_{k+i} u_\triangle(t)$ is the triangular waveform with the predicted peak current value of $\hat{I}_{k+i}$.

5. OPTIMUM FREQUENCY SELECTION

The Fourier transformation of current load (15) can be written as

$$H(j\omega) = \sum_{i=0}^{K} H_i(j\omega) e^{-ji\omega T}, \tag{16}$$

where $H_i(j\omega)$ $(i > 0)$ is the Fourier transformation of $u_i(t)$, and $H_0(j\omega)$ is the Fourier transformation of $u(t)$ for $t \le 0$. According to the discussion as shown in section 2, our goal is to minimize the resonance noise, i.e., the magnitude of the frequency domain response of current load $H(j\omega)$, at $\omega = \omega_0$, i.e.,

$$\min_T \left| \sum_{i=0}^{K} H_i(j\omega_0) e^{-ji\omega_0 T} \right| + \lambda(T - T_{min}), \tag{17}$$

where T_{min} is the clock period corresponding to the maximum clock frequency. The reason for us to add a weighted penalty function $\lambda(T - T_{min})$ to the objective function is to consider the impact of performance loss resulting from changing clock frequency. The positive number λ reflects aggressiveness of our frequency actuator design.

It is clear that (17) is an unconstrained nonlinear optimization problem and any general optimization techniques such as Newton's method can be applied to solve it efficiently.

In practice, by knowing the fact that only a finite number of discrete clock frequencies are available for any digital-based programmable PLL design, we develop a more efficient way of solving the problem. We denote the finite set of available programmable frequencies as $\{1/T_1, \ldots, 1/T_q, \ldots, 1/T_m\}$, then we can easily find the optimal frequency by evaluating (17) over different T_q and select the optimal one that minimizes the objective function, i.e., (17) can be rewritten as

$$\min_{T_q} \left| H_0(j\omega_0) + \sum_{i=1}^{K} \hat{I}_{k+i} H_\triangle(j\omega_0) e^{-ji\omega_0 T_q} \right| + \lambda(T_q - T_{min}), \tag{18}$$

where $H_\triangle(j\omega)$ is the Fourier transformation of the unit triangular waveform $u_\triangle(t)$. This is exactly the optimization problem we need to solve at clock cycle k. To further improve efficiency in evaluating (18), we can pre-calculate and store

$$A_{i,q} = H_\triangle(j\omega_0) e^{-ji\omega_0 T_q} \tag{19}$$

in a look-up table, as $A_{i,q}$ is a floating number for $1 \le i \le L$ and $1 \le q \le m$.

6. EXPERIMENTAL RESULTS

6.1 Current Prediction Verification

We first verify the accuracy and efficiency of our prediction algorithm with current data measured on a mobile chip from industrial design. We apply both the predetermined linear filter and the LMS adaptive filter designs to our frequency actuator, and the prediction results based on simulation are illustrated in Fig. 4 (a) and (b), respectively. Both methods use 32 points ($M = 32$) in history with spacing $L = 400$, and predict the currents in the incoming 400 clock cycles. From the figure we can quantitatively see that the adaptive filter can provide a better prediction result (closer to the actual current) than the predetermined linear filter. Experimental results show that adaptive filter has an average prediction error of 1.51%, whereas that of the predetermined linear filter is 13.4%. On the other hand, we observe that the maximum prediction error for adaptive filter can be as large as 311% in time period $5650 - 5700ns$, indicating the failure of convergence, whereas the predetermined linear filter has an error of 11.6% in those clock cycles. Fortunately, such error does not affect the proposed resonance reduction as explained below.

Fig. 5 illustrates the predicted current spectrum from predetermined filter (a) and adaptive filter (b) compared with the actual current spectrum. From the figure we can see that both methods are accurate when the responses are sharp. The main prediction error only happens at frequencies where the frequency domain response is small, and thus does not affect our selection of correct clock frequency.

Fig. 6 (a) shows the relationship between the average prediction error and the number of history data points M for fixed spacing $L = 400$. We can see that the prediction accuracy improves with the increase of M. For the region of $M < 30$, increasing M can result in a big decrease in the average error, while for the region of $M > 30$, changing M has little impact on the error. Fig. 6 (b) shows the relationship between the average prediction error and the spacing

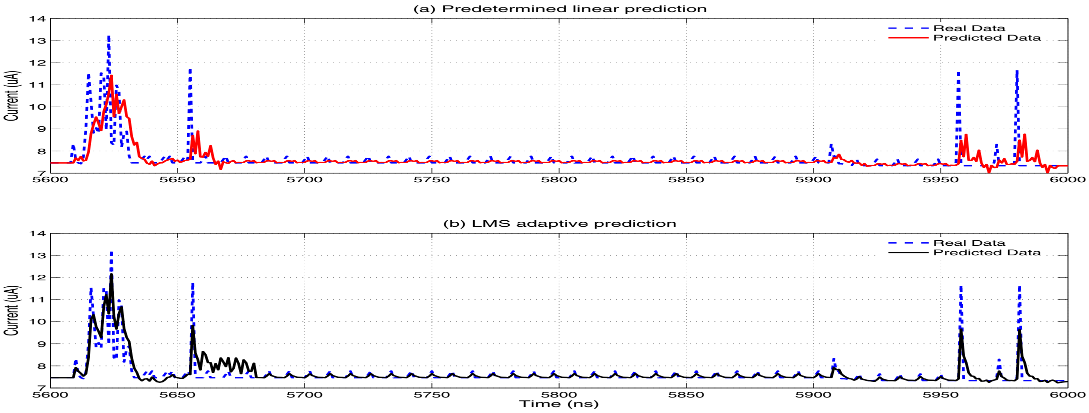

Figure 4: The current prediction results.

Table 1: Noise comparison between three models.

design	max noise (V)				average noise (V)			
	Baseline	Retroactive	Proactive		Baseline	Retroactive	Proactive	
			Predetermined	Active			Predetermined	Active
mobile	0.33	0.29 (-12%)	0.14 (-58%)	0.07 (-79%)	0.14	0.11 (-21%)	0.08 (-43%)	0.03 (-79%)
μP	0.36	0.31 (-14%)	0.14 (-61%)	0.08 (-78%)	0.15	0.10 (-33%)	0.05 (-67%)	0.02 (-87%)

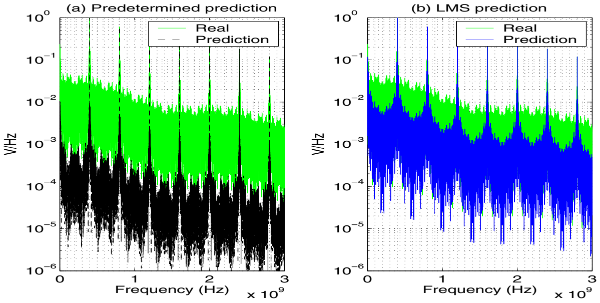

Figure 5: Comparison of spectrum results.

Table 2: Normalized system latency overhead comparison between three models.

Design	Normalized system latency overhead			
	Baseline	Retroactive	Proactive	
			Predetermined	Active
mobile	0.27	0.21 (-22%)	0.07 (-74%)	0.02 (-93%)
uP	0.31	0.20 (-35%)	0.11 (-65%)	0.05 (-84%)

L for fixed number of history data points $M = 32$. From the figure, we see that there is an optimal L that corresponds to the smallest error for both methods, and such an optimal value is roughly the same ($L = 400$) for both methods. Another interesting observation is that the adaptive filter is less sensitive to parameter changes than the predetermined filter. This is expected as the adaptive property enables it to adjust itself with the change of parameters.

6.2 Resonance Noise Reduction

We first study how the number of current sensors affects noise reduction on the same mobile chip. As shown in Fig. 7, the noise reduction is almost the same when the number of current sensors is greater than 5% of the total number of system ports, which translates to $10 - 100$ current sensors for a leading chip. This suggests that there is no need to place many sensors for the measurement. Next we conduct experiments for the same mobile chip and one additional high performance micro-processor to illustrate the impact on resonance noise reduction. We assume that the current profile obtained from measurement scales with the clock cycle. For these two designs, the tracking time for PLL is 75 clock cycles. The choice of clock frequencies ranges from 1.5 GHz to 0.8 GHz with an interval of 0.1 GHz. The retroactive model incrementally reduces the clock frequency by 0.1 GHz until the noise is below the tolerance bound. Then it tries to incrementally increase the clock frequency with 0.1 GHz step until the maximum frequency is reached or when noise violation occurs. The proactive model select optimal frequency based on predicted currents. We apply simulation with the current profile and the distributed PDN to get the maximum and average voltage droop. The comparison results are shown in Table 1. Compared with the baseline model without frequency actuator, the retroactive approach can only reduce the max noise by up to 14% and reduce the mean noise by up to 33%. On the other hand, our proactive approach with predetermined linear filter can reduce the max noise by up to 61% and the mean noise by up to 67%, while the proactive approach with the LMS adaptive filter can reduce the max noise by up to 79% and the mean noise by up to 87%.

Table 3: Normalized gate count comparison between three models from Cadence Encounter RTL complier.

Design	Normalized gate count		
	Baseline	Proactive	
		Predetermined	Active
mobile	1	1.0005	1.002
uP	1	1.0002	1.004

6.3 Performance and Area Impact

For both the microprocessor and mobile examples, we simulate the system latency for one time resonance noise violation such that one time reboot is required in the baseline case. From the design, it takes $1\mu s$ to do a full save and $1\mu s$ to do a restore of the whole architecture state. The ideal latencies for the retroactive and proactive approaches are, respectively, $8M$ and $20M$ cycles. The latency overhead includes time of potential reboot, time of clock frequency

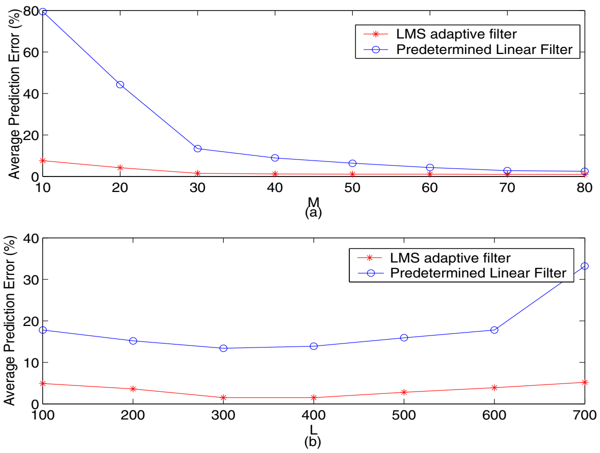

Figure 6: The average prediction error for both methods w.r.t (a) the number of history data points (M) and (b) their spacing (L).

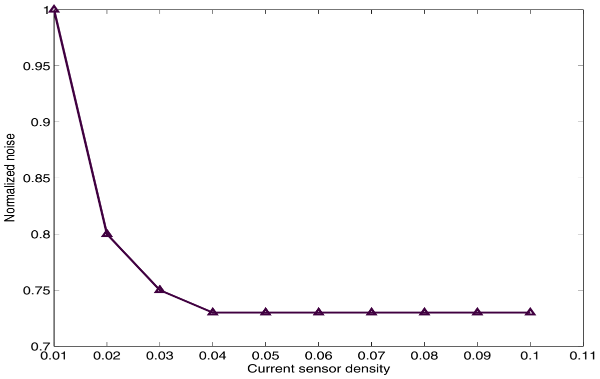

Figure 7: Normalized noise vs. current sensor density.

switches to avoid resonance noise and to increase clock frequency when the resonance is gone, and time loss due to slowing down the clock. In Table 2, we report normalized latency overhead with respect to the ideal latencies for the baseline, retroactive and proactive cases. For proactive case, we have tested both the predetermined linear filter and the LMS adaptive filter. From the table, we see that compared with the baseline model, the retroactive method can reduce the system latency overhead by up to 35%, the proactive model with the predetermined linear filter can reduce that by up to 74%, and the proactive model with the LMS adaptive filter can reduce that by up to 93%. This further illustrates the importance of the proactive frequency actuator for high performance systems.

We also compare the gate count for the LMS adaptive filter and the predetermined linear filter based designs obtained from Cadence Encounter RTL Compiler, and the results are reported in Table 3. From the table, we see that the predetermined linear filter based actuator can cause the gate count to be increased only by 0.02% for the microprocessor design, while the design of the LMS adaptive filter causes the gate count to increase by 0.4%.

7. CONCLUSIONS

Because of the distributed RLC characteristics of a power delivery network (PDN), runtime resonance noise at the low-to-middle frequency range may significantly affect the reliability of a PDN and chip performance. In contrast to existing retroactive solution that only remedies the noise problem when the noise problem has occurred already, we have proposed a novel design approach to proactively suppress resonance noise. We have developed an efficient stochastic current load prediction method based on a generalized Markov process modeling. We have presented a frequency actuator that utilizes both on-chip dynamic current sensors and a programmable PLL for frequency adjustment. A novel optimal frequency selection algorithm has also been developed. Compared with baseline design without frequency actuator, experimental results show that our frequency actuator design alone reduces maximum noise by 16% and average noise by 30%, while our proactive frequency actuator with current prediction reduces maximum noise by 77% and average noise by 85%. In terms of system level performance, compared with the baseline model, our frequency actuator alone can reduce the system latency overhead by up to 35% , and with current prediction it can reduce that by up to 93%.

8. REFERENCES

[1] K. Bathey and M. Swaminathan, “Resonance analysis and simulation in packages,” in *Electrical Performance of Electronic Packaging*, pp. 169–172, Oct. 1995.

[2] H. Su, S. S. Sapatnekar, and S. R. Nassif, “Optimal decoupling capacitor sizing and placement for standard-cell layout designs,” *IEEE Trans. on CAD*, vol. 22, pp. 428–436, April 2003.

[3] K.-H. Erhard, F. Johannes, and R. Dachauer, “Topology optimization techniques for power/ground networks in VLSI,” in *IEEE/ACM DATE*, 1992.

[4] X. D. Tan and C. J. Shi, “Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings,” in *IEEE/ACM DAC*, pp. 78–83, 1999.

[5] M. Zhao, R. Panda, S. Sundareswaran, S. Yan, and Y. Fu, “A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming,” in *IEEE/ACM DAC*, 2006.

[6] Y. Shi, J. Xiong, C. C. Liu, and L. He, “Efficient Decoupling Capacitance Budgeting Considering Operation and Process Variations,” in *IEEE/ACM ICCAD*, 2007.

[7] J. Xu, P. Hazucha, M. Huang, P. Aseron, F. Paillet, G. Schrom, J. Tschanz, C. Zhao, V. De, T. Karnik, and G. Taylor, “On-Die Supply-Resonance Suppression Using Band-Limited Active Damping,” in *IEEE International Solid State Circuits Conference*, pp. 286–288, 2007.

[8] M. S. Gupta, J. L. Oatley, R. Joseph, G. Wei, and D. Brooks, “Understanding Voltage Variations in Chip Multiprocessors using a Distributed Power-Delivery Network,” in *IEEE/ACM DATE*, 2006.

[9] M. Ang, R. Salem, and A. Taylor, “An On-Chip Voltage Regulator Using Switched Decoupling Capacitors,” in *IEEE International Solid State Circuits Conference*, 2000.

[10] J. Liu and R. Makki, “SRAM Test Using On-chip Dynamic Power Supply Current Sensor,” in *International Workshop on Memory Technology, Design and Testing*, 1998.

[11] Y. Lechuga, R. Mozuelos, M. Martinez, and S. Bracho, “Built-in Dynamic Current Sensor for Hard-to-Detect Faults in Mixed-Signal ICs,” in *IEEE/ACM DATE*, 2002.

[12] A. Waizman and C.-Y. Chung, “Resonant Free Power Network Design Using Extended Adaptive Voltage Positioning (EVAP) Methodology,” *IEEE Trans. on Advanced Packaging*, 2001.

[13] Y. Sumi and et al, “PLL frequency synthesizer with an auxiliary programmable divider,” in *IEEE ISCAS*, pp. 532–536, Jul. 1999.

[14] S. Khadanga, “Synchronous programmable divider design for PLL Using 0.18 um CMOS technology,” in *International Workshop on System-on-Chip for Real-Time Applications*, 2003.

[15] A. K. Basu, *An Introduction to Stochastic Process*. Alpha Science Int'l Ltd., 2003.

[16] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2002.

[17] D. Kouroussis, I. A. Ferzli, and F. N. Najm, “Incremental partitioning-based vectorless power grid verification,” in *IEEE/ACM ICCAD*, 2005.