

Decoupling Capacitance Allocation for Timing with Statistical Noise Model and Timing Analysis

Takashi Enami[†] Masanori Hashimoto[†] Takashi Sato[‡]

[†]Dept. Information Systems Engineering, Osaka University
1-5 Yamadaoka, Suita, Osaka, Japan

Email: {enami.takashi, hasimoto}@ist.osaka-u.ac.jp

[‡]Integrated Research Institute, Tokyo Institute of Technology
4259-R2-17, Nagatsuta, Midori-ku, Yokohama, Japan

Email: sato@lsi.pi.titech.ac.jp

Abstract— This paper presents an allocation method of decoupling capacitance that explicitly considers timing. We have found and focused that decap does not necessarily improve a gate delay at all the switching timing within a cycle, and devised an efficient sensitivity calculation of timing to decap for decap allocation. The proposed method, which is based on a statistical noise modeling and timing analysis, accelerates the sensitivity calculation with an approximation and adjoint sensitivity analysis. Experimental results show that the decap allocation based on the sensitivity analysis efficiently optimizes the worst-case circuit delay within a given decap budget. Compared to the maximum decap placement, the delay improvement due to decap increases by 5% even while the total amount of decap is reduced to 40%.

I. INTRODUCTION

Power supply noise has become one of primary concerns in modern high performance circuit design due to increased current consumption and lowered supply voltage. Power supply noise mainly consists of IR drop and Ldi/dt noise. Widening power wire is a common technique to reduce IR drop with a sacrifice of routability and wire resource. However, it cannot reduce Ldi/dt noise originating from package and bonding wires. In recent designs, decoupling capacitance (decap) has been placed in power supply network to suppress IR drop and Ldi/dt noise. Decap is often implemented with MOS gate capacitance, and a large decap consumes large silicon area. Moreover, gate leakage current has increased along miniaturization of transistor, and hence it becomes harder to place a large amount of decap within a given leakage constraint. Therefore, necessary and sufficient decaps should be placed at appropriate positions.

Power supply noise is becoming more and more influential on timing because of lowered supply voltage, and hence power supply network alleviating the noise effect on timing is highly demanded. However, most of conventional design techniques aim to reduce power supply noise, and not to directly minimize the impact on timing. In the past, efficient allocation of decaps has been studied [1], [2], [3]. References [1], [2] proposed methods to reduce voltage drop with a limited amount of decap. In Refs. [1], [2], an allowable voltage drop is first specified, and the integration of the excessive drop with respect to time is minimized within the given decap budget. However, the suppression of voltage drop does not necessarily improve timing as reported in Ref. [4]. To minimize silicon area and leakage of decaps and maximize the timing improvement thanks to decap, the timing should be directly and explicitly considered in decap design.

Reference [3] proposed a decap allocation method to improve timing within a decap budget. The authors compute

sensitivities of an objective function that mainly consists of timing variation using adjoint sensitivity analysis, and find positions suitable for decap placement. However, a problem is that, at each position, the worst voltage drop within a specified clock cycle is computed and used for gate delay calculation without considering switching timing window. Decap nicely reduces peak voltage drop, but does not improve supply voltage at all the timing within a clock cycle, which will be illustrated in Section II. Reference [3] assumes that reduction of peak voltage drop necessarily improve timing, but it is not *true*. Moreover, a noise waveform changes cycle by cycle, and the noise reduction effect of decap also varies. Spatial noise variation and within-cycle and inter-cycle temporal noise variation caused by decap insertion/removal must be considered in timing-oriented decap allocation.

In this paper, we propose a decap allocation method for timing that explicitly considers spatial and temporal variation of noise due to decap modification. To consider noise dependence on input patterns, we model dynamic power supply noise and its variation due to decap modification statistically, and compute statistical sensitivity. For efficient sensitivity computation, we devise a performance function, which can be efficiently computed with an approximation, for adjoint sensitivity analysis [7]. Guided with the computed sensitivity, we identify decap positions that improve timing.

This paper is organized as follows. Section II illustrates how decaps affect timing and demonstrates that decap does not necessarily improve timing. Section 3 describes the proposed decap allocation method. Section 4 demonstrates that the proposed method can improve timing via decap allocation, and finally Section 5 concludes the discussion.

II. DECAP EFFECT ON TIMING

Decoupling capacitance suppresses the peak voltage drop, however the average voltage within a clock cycle hardly improves. Decaps are discharged to supply charge to current hungry nodes as temporal current sources. After that, decaps must be recharged. When the decap is large, a larger amount of charge must be restored, which means the charge time for recovery, which is tightly related to RC time constant, becomes longer. This phenomenon is illustrated in Fig. 1. After a decap addition, the peak voltage drop is significantly reduced. In contrast, the supply voltage in the latter half of the clock cycle is lower than that before the decap addition. This means that decap does not necessarily improve supply voltage at all the timing within a cycle. At some timings, the supply voltage improves, which results in timing improvement. On the other

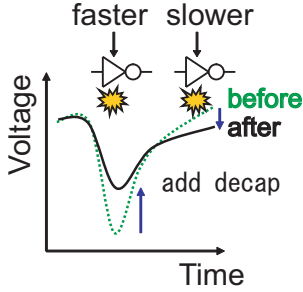


Fig. 1. Decap effect on timing.

hand, at the other timings, the supply voltage worsens and gate switching delay increases. Suppose the left gate in Fig. 1 is in a critical path. In this case, the critical path delay is improved by the decap addition. In contrast, the circuit delay increases in the case that the right gate of Fig. 1 belongs to a critical path. Furthermore, in this case, decreasing decap may improve the critical path delay, which might not be widely known in designers. Without considering such attribute of decaps, the timing can not be efficiently improved via decap allocation.

The effect of decaps on timing improvement depends on the position of a critical path, and hence appropriate positions that maximize timing improvement should be selected. We here show impacts of decap position on the worst-case delay as an example. In this paper, the circuit delay is estimated by a noise aware SSTA [5], which gives a circuit delay distribution considering noise dependence on input patterns. We define the worst-case delay as $\mu + 3\sigma$ for a case, while other definitions delay also can be used, where μ and σ are the average and the standard deviation of the circuit delay, respectively. In this example, a power and ground network (Fig. 2) was attached to a divider circuit [6], where both power and ground networks are separately modeled though the figure is simplified. In this circuit, a 100fF capacitance is initially placed between power and ground at each node. We then calculated the worst-case delay of c6288 (one of ISCAS85 benchmark circuits) when the divider noise is given. We evaluated the variation of the worst-case delay in the case that a 3pF decap is added into each power/ground node. Table I lists the worst-case delay variation. When a decap is placed at (0, 3) or (1, 3), the worst-case delay is improved by more than 5ps, whereas six positions increase the circuit delay. In this example, we should place decaps at (0, 3) and (1, 3), and must not place decaps at (2, 0) and (2, 1).

To identify positions where decap should be increased, we need to compute sensitivities of timing at each candidate of decap placement. In the next section, we discuss how to efficiently compute the sensitivities and how to assign decaps.

III. PROPOSED DECOUPLING CAPACITANCE ALLOCATION

A. Problem formulation and overview of proposed method

This paper discusses a problem formulated such that the amount of decap at each placeable position in a circuit is adjusted from a initial layout to minimize the worst-case delay using a given amount of decaps. At each placeable position, the maximum decap size is specified.

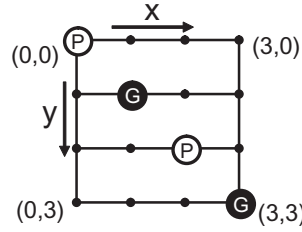


Fig. 2. Power/ground network for evaluating position dependence of decap on timing.

TABLE I
POSITION DEPENDENCE OF DECAP ON TIMING IMPROVEMENT.

initial worst-case delay: 2882ps			
pos. (x,y)	imp. (ps)	pos. (x,y)	imp. (ps)
(0,0)	0.241	(2,0)	-1.84
(0,1)	1.11	(2,1)	-2.08
(0,2)	4.34	(2,2)	0.192
(0,3)	7.27	(2,3)	4.70
(1,0)	-0.579	(3,0)	-1.79
(1,1)	-0.142	(3,1)	-1.24
(1,2)	0.809	(3,2)	0.224
(1,3)	9.26	(3,3)	1.12

The proposed method first calculates the sensitivity of circuit delay to decap at each placeable position. When the amount of decap is varied, power supply noise changes, which results in timing variation. We therefore express the sensitivity as the product of two sensitivities; sensitivity of power supply noise to decap and sensitivity of timing to power supply noise. Once one capacitance value is changed, its effect spreads spatially in the circuit, and the voltages at many points vary. We thus define the sensitivity of circuit delay to i -th decap C_i as follows.

$$\frac{\partial Delay}{\partial C_i} = \sum_{\forall j} \left(\frac{\partial Delay}{\partial V_j} \times \frac{\partial V_j}{\partial C_i} \right), \quad (1)$$

where $Delay$ is the circuit delay, and V_j is the voltage at j -th point.

Section 3.2 explains a statistical noise modeling used in the sensitivity calculation and timing analysis. Section 3.3 presents efficient computation of $\frac{\partial Delay}{\partial C_i}$. Finally section 3.4 describes decap allocation based on the sensitivities.

B. Statistical modeling of power supply noise

We first introduce a statistical model of power supply noise proposed in Ref. [5]. Noise waveforms differ cycle by cycle depending on input patterns. Power supply noise varies continuously in space and time, and strictly speaking, every cell has different noise waveform. We spatially and temporally discretize power supply noise and assign random variables. We then compute statistical properties, such as average, standard deviation and correlation, of the assigned random variables.

We first determine voltage observation points by spatially discretizing a chip. The spatial discretization is performed by partitioning a chip/block area into a 2D grid and choosing a representative value for each divided partition. As a representative value, for example, the voltage at the center point (Fig. 3) or the average voltage in each partition is a candidate. The voltages of all nodes in the same partition are assumed to be identical.

An important property of power supply noise is its dynamic behavior. To express dynamic waveforms within a cycle, we partition a clock cycle into several time spans, and compute a representative voltage (e.g. average as shown in Fig. 4).

We then assign a random variable to power supply or ground voltage at each time span and at each spatial grid. We hereafter this assigned random variable as a power variable. We treat the voltage value at every clock cycle as a different sample. Figure 4 shows an example when the voltage at position (x, y)

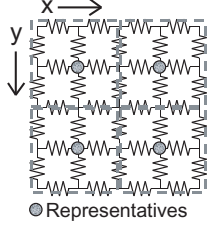


Fig. 3. Spatial discretization. Divided into partitions with broken lines.

is divided into three time spans and its random variables are denoted as $V_{x,y,1}$, $V_{x,y,2}$ and $V_{x,y,3}$. The number of time spans is determined according to the modeling requirement, i.e. when we need to accurately model dynamic variation within a clock cycle, the number of spans should be increased, otherwise several spans are sufficient.

To efficiently perform statistical timing analysis, Ref. [5] orthogonalizes the variables with principal component analysis, and derives a statistical model including the statistical information such as averages, standard deviations and correlation coefficients of the variables.

C. Sensitivity calculation of circuit delay to decap

In this section, we propose an efficient way to calculate Eq. (1). The sensitivity of circuit delay to a power variable $\frac{\partial Delay}{\partial V_j}$ can be computed by SSTA considering power supply noise [5]. The sensitivity of a power variable to a decap $\frac{\partial V_j}{\partial C_i}$ can be calculated with a circuit simulation. However, both of them require a large number of SSTA and circuit simulation. SSTA must be repeated for the number of power variables to obtain all the sensitivities. Circuit simulation must be performed for the number of decaps. Therefore, the required cost for the sensitivity computation is prohibitively expensive. We therefore compute $\frac{\partial Delay}{\partial V_j}$ with an approximation, and devise a performance function directly corresponding the circuit delay for adjoint sensitivity analysis [7] to calculate $\frac{\partial Delay}{\partial C_i}$ efficiently. The followings explain how to compute $\frac{\partial Delay}{\partial V_j}$ and $\frac{\partial Delay}{\partial C_i}$ in detail.

1) *Sensitivity calculation of $\frac{\partial Delay}{\partial V_j}$* : We first calculate the sensitivity of circuit delay to power variable $\frac{\partial Delay}{\partial V_j}$. Every gate delay is varied by voltage fluctuation, but the delay variation of each gate does not necessarily affect the circuit delay, because the gate with a large timing slack has little influence on the circuit delay. We therefore have to consider how much each gate delay variation contributes to the circuit delay variation. To do this, we calculate ‘‘criticality’’ defined as a probability that a gate belongs to the critical path of a circuit. Several methods to compute criticality have been proposed [8], [9]. We adopted the method proposed in Ref. [8] in this paper for the sake of implementation simplicity, though other methods also can be used. Criticality is computed in two stages. The first stage is tightness probability calculation at every gate, where the tightness probability is a probability that an input is selected in MAX operation for the latest arrival time computation at the gate, and is computed for every input. This calculation is already performed inside SSTA, and hence no

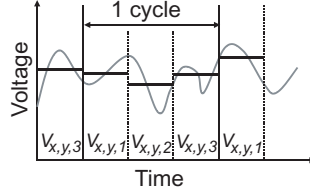


Fig. 4. Temporal discretization. Dividing a clock cycle into time spans.

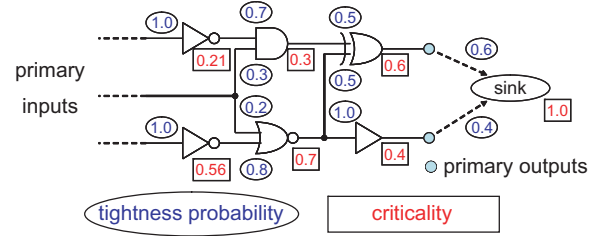


Fig. 5. Criticality computation.

additional computational cost is necessary. The second stage is the criticality computation. We illustrate the computational flow in Fig. 5. First, the criticality of the (virtual) sink node is set to 1, because the sink node is always included in the critical path. Then, the criticality of each gate is calculated with a backward traversal of the timing graph.

$$Cr_i = \sum_{j \in \text{outputs}(i)} Cr_j \times Tp_{j,i}, \quad (2)$$

where Cr_i is criticality of gate i , Cr_j is criticality of fan-out gate j , and $Tp_{j,i}$ is the tightness probability to select the arrival time of gate i at gate j .

We then approximate the sensitivity of a circuit delay to a power variable $\frac{\partial Delay}{\partial V_j}$ using criticality. Focusing on a single gate, the contribution of the gate to the circuit delay variation depends on not only criticality but also on the sensitivity of the gate delay to power supply voltage. We thus express the contribution of the gate as the product of these two factors. We next focus on power variables. A power variable has a strong impact on circuit delay when many gates that have large contributions on circuit delay are associated with the power variable. Considering a fact that voltage variation both at a receiver and at drivers affects gate delay, the sensitivity of circuit delay to power variable $V_{x,y,t}$ is described as

$$\begin{aligned} \frac{\partial Delay}{\partial V_{x,y,t}} = & \sum_{i \in (x,y,t)} \left[Cr_i \sum_{j \in \text{inputs}(i)} \left(\left(\frac{\partial d_i}{\partial V_{re}} \right)_j \cdot Tp_{j,i} \right) \right. \\ & \left. + \sum_{k \in \text{outputs}(i)} \left\{ \sum_l \left(Cr_k \cdot \left(\frac{\partial d_k}{\partial V_{dr_i}} \right)_l \cdot Tp_{k,l} \right) \right\} \right], \quad (3) \\ & [\text{where } l \in (\text{inputs}(k) \in (x', y', t))], \end{aligned}$$

where $\left(\frac{\partial d_i}{\partial V_{re}} \right)_j$ is the sensitivity of gate delay d_i from input j to power variable V_{re} . The former term represents the contribution of gate i as a receiver, and the latter term is the contribution as drivers. Gate i also contributes the latter term when another gate l at position (x', y') drives gate k at time span t . Here, tightness probability is used to exclude the contribution of non-critical gates.

We thus calculate $\frac{\partial Delay}{\partial V_j}$ in Eq. (1) using Eq. (3). This approximation will be experimentally validated in Section 4.2. With this approximation, we do not have to repeat SSTA for the number of power variables.

2) *Sensitivity calculation of $\frac{\partial Delay}{\partial C_i}$ using adjoint sensitivity analysis*: We next discuss how to compute $\frac{\partial Delay}{\partial C_i}$ using $\frac{\partial Delay}{\partial V_j}$. A straightforward way to obtain $\frac{\partial Delay}{\partial C_i}$ in Eq. (1)

is to compute $\frac{\partial V_i}{\partial C_i}$ by changing C_i slightly and simulating power supply noise. A problem is that we have to perform the simulation for all the decaps, which is impractical in computational cost. To solve this problem, we use adjoint sensitivity analysis. By appropriately setting a performance function used in adjoint sensitivity analysis, we can directly obtain $\frac{\partial Delay}{\partial C_i}$ instead of $\frac{\partial V_i}{\partial C_i}$. We here suppose that power distribution network is modeled as a linear circuit and switching gates are expressed as current sources.

Adjoint sensitivity analysis simulates two circuits; an original circuit and its adjoint circuit. The adjoint circuit is constructed as follows. First, all independent voltage and current sources are removed from the original circuit. Then, current sources are inserted to voltage observation points, and their current are given based on a performance function of time τ , $f(\tau)$. We finally want to obtain $\frac{\partial Delay}{\partial C_i}$ in Eq. (3), and hence $\int_0^T f(\tau) d\tau$ should be a circuit delay function w.r.t. power variables.

$$\begin{aligned} Delay &= Delay_{init} + \sum_x \sum_y \sum_t \frac{\partial Delay}{\partial V_{x,y,t}} (V_{x,y,t} - V_{x,y,t}^{init}), (4) \\ &= \sum_{\tau=0}^T f(\tau) \left(= \int_0^T f(\tau) d\tau \right), (5) \end{aligned}$$

where the time is divided into several time spans as mentioned in Section 3.2, and hence the integral is expressed as a summation. Thus, the performance function $f(\tau)$ is expressed with power supply (ground) voltage of node n ($V_n(\tau)$) as

$$f(\tau) = D(\tau) + \left[\sum_x \sum_y \left\{ W_{x,y}(\tau) \sum_{n \in (x,y)} V_n(\tau) \right\} \right]. (6)$$

Here, $D(\tau)$ is the term independent of $V_n(\tau)$, and $W_{x,y}(\tau)$ is $\frac{\partial Delay}{\partial V_j}$. If the number of temporal division used in the statistical noise modeling is 3, $W_{x,y}(\tau)$ is also discretized and described as

$$W_{x,y}(\tau) = \begin{cases} \frac{\partial Delay}{\partial V_{x,y,1}} & (0 \leq \tau < bt_{1,2}) \\ \frac{\partial Delay}{\partial V_{x,y,2}} & (bt_{1,2} \leq \tau < bt_{2,3}) \\ \frac{\partial Delay}{\partial V_{x,y,3}} & (bt_{2,3} \leq \tau < T). \end{cases} (7)$$

where $bt_{i,j}$ is the boundary time between span i and span j , and T is the clock period. Strictly speaking, in some cases, $W_{x,y}(\tau)$ must be divided by a integer depending on the simulation setup and the variable assignment.

The current of the current source attached to node n , Φ_n , is represented as

$$\begin{aligned} \Phi_n(\zeta = (T - \tau)) &= -\frac{\partial f(\tau)}{\partial V_n}, \\ &= -W_{x,y}(\tau). \end{aligned} (8)$$

Now that we can obtain the simulation result of the adjoint circuit, we next calculate the sensitivity of circuit delay to capacitance C_i by convolution.

$$\frac{\partial Delay}{\partial C_i} = - \int_0^T \{ \psi_{C_i}(T - \tau) v_{\dot{C}_i}(\tau) \} d\tau, (9)$$

where ψ_{C_i} is the voltage difference of C_i in the adjoint circuit, and $v_{\dot{C}_i}$ is the derivative of the voltage difference of C_i in the original circuit.

We thus obtain the sensitivity of circuit delay to power variable. Please note that though $\frac{\partial Delay}{\partial V_j}$ includes an approximation, the other computation is exact thanks to adjoint sensitivity analysis. With a single adjoint network analysis, we can obtain the sensitivities to all the decaps.

D. Decap allocation

We explain how to determine decaps based on the sensitivities. We sort the decaps based on the sensitivities, and assign decaps in the order.

Noise variation due to decaps varies cycle by cycle, because current consumption fluctuates according to input patterns. It is difficult to determine decaps just looking at a particular clock cycle. We calculate the sensitivities for a certain amount of cycles, and obtain the sensitivity distribution. Here, we already have noise waveforms for a certain amount cycles used in the statistical noise modeling, and then it is easy to get the sensitivity distribution. The sensitivity of C_i in m th clock cycle is calculated by

$$\frac{\partial Delay}{\partial C_i} = - \int_0^T \{ \psi_{C_i}(T - \tau) v_{\dot{C}_i}((m-1)T + \tau) \} d\tau. (10)$$

We have several choices in how to sort the decaps, since each sensitivity has a distribution. We have tried some choices and empirically found that the ordering based on the average gave good optimization results. Thus, we sort the decaps using the average sensitivity.

After the sorting, we assign the size of each decap. When the timing varies nonlinearly w.r.t. the decap size, we have to carefully choose the decap size. On the other hand, when the change of decap size hardly varies the sensitivities, it is reasonable to assign the maximum size to the decap with a good sensitivity. We have experimentally verified that the sensitivities did not change significantly even after the decap allocation, which will be shown in Section 4.3. We therefore first set all the decap sizes to zero, and assign the maximum decap size from the top in the sensitivity order until the decap budget is fully used. The algorithm is very simple, but it efficiently works, which will be shown in Section 4.4.

IV. EXPERIMENTAL RESULTS

This section demonstrates experimental results. We first validate the sensitivity analysis, and then show that the proposed method can improve timing.

A. Experimental conditions

We here explain experimental conditions. For the simplicity of the implementation, delay times of ISCAS85 benchmark circuits, a 64-bit multiplier, and an ALU circuit for vector operation were analyzed under the power supply noise of an FPU circuit [6]. Here, the noise generator and the timing-analyzed circuits were different for just the simplicity of the implementation, and there is no technical difficulty to analyze

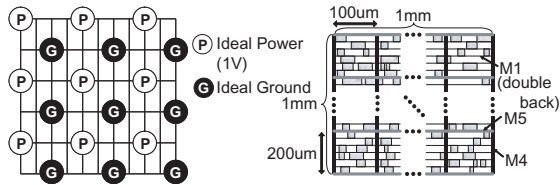


Fig. 6. Power and ground network of test circuit.

the timing of the noise generator circuit. Both circuits were synthesized by a commercial logic synthesizer and placed and routed by a commercial tool with a 90nm standard cell library. We attached a power/ground network shown in Fig. 6 to the noise generator circuit and simulated the power supply noise with input vectors of 2000 clock cycles. We supposed that this circuit was designed with an SOI technology. In statistical modeling of power supply noise, the numbers of spatial and temporal division were set to 10×10 and 10, respectively. Decoupling capacitance per area was calculated assuming decap was designed with $1 \mu\text{m}$ gate length in the 90nm process. Decaps could be placed up to 50% of the circuit area at every point, for simplicity. The number of points was set to 3960. We calculated the sensitivities to decaps in the case that 25% circuit area was occupied with decaps, i.e. 50% of the maximum decaps were inserted.

B. Validation of sensitivity calculation

We first validate the sensitivity calculation discussed in Section 3. $\frac{\partial \text{Delay}}{\partial V_j}$ computation introduces an approximation using criticality and tightness probability. The other point of the proposed sensitivity computation is exact thanks to adjoint sensitivity analysis. Therefore, we here verify the estimation accuracy of $\frac{\partial \text{Delay}}{\partial V_j}$. For a comparison, we also calculated the improvement of the worst delay in the case that the average of one power variable is improved by 1mV. When the power variable corresponds to power/ground, the average increases/decreases by 1mV. The sensitivities estimated by this exact procedure and the proposed method are plotted in Fig. 7, and each dot corresponds to a power variable. We can see that the sensitivity is well estimated with Eq. (3), though the computational cost of the proposed method is negligibly small. The exact method has to perform SSTA for the number of power variables, whereas the proposed method requires a simple backward trace of the timing graph only once for criticality computation.

C. Sensitivity variation due to decap allocation

The proposed method changes many decaps simultaneously based on the sensitivities evaluated at once. Generally speaking, when a large variation occurs in a power distribution network, the sensitivities could significantly change. If the sensitivities are totally different before and after decap allocation, we have to incrementally update the sensitivities. We here evaluate how different the sensitivities are before and after decap allocation.

We suppose two circuits; an initial circuit and a circuit optimized by the proposed method. Both circuits have the same amount of decap in total and it corresponds to 25%

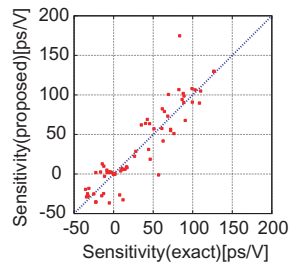


Fig. 7. Accuracy verification of sensitivity computation of $\frac{\partial \text{Delay}}{\partial V_j}$ (c432).

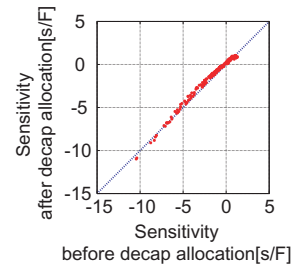


Fig. 8. Sensitivity comparison before and after decap allocation (ALU).

of the total circuit area. In the optimization, we selected top 50% of decap positions in the order of the sensitivity, and increased their sizes to the maximum, and at other positions, the decaps are removed. In the initial circuit, decaps are uniformly distributed in space. Figure 8 shows the sensitivities before and after the decap allocation. The sensitivities are almost unchanged, which means it is hardly necessary to incrementally update the sensitivity. Figure 8 also tells us that even though the decap size is changed to the maximum or zero, the sensitivities are still acceptably accurate, and hence an intermediate size between zero and the maximum does not have to be selected.

D. Timing optimization via decap allocation

We finally demonstrate timing optimization results by the proposed method under various conditions of decap budget. To reveal the effectiveness of the proposed method, we compare the worst-case timing of two circuits. In the first circuit, decaps are uniformly placed in the circuit. The second circuit was optimized by the proposed method. In both circuits, the total amount of decaps is the same. Figures 9 and 10 show the worst-case delays of c432 and c1908 circuits with various decap budgets. In horizontal axis, 50% of decap rate means that 50% of the circuit area is occupied by the decap. In Fig. 9, increasing the amount of decaps did not always improve the delay. Moreover, Fig. 10 shows that increasing the amount of decaps deteriorates the delay of c1908 circuit. Both in Figs. 9 and 10, the decaps allocated by the proposed method works better than the uniformly placed decaps. In these experiments, the same power supply noise are given to every circuit. However, the effect of decaps on timing varies, which means the optimal decap allocation is strongly dependent on circuits and explicit consideration of timing is necessary in decap design.

Table II shows the worst-case delays of the delay-optimized circuits, those without decaps and those with the maximum decaps. The third column represents the delay with ideal power supply voltage. The fourth column represents the delays in the case that no decap is allocated. The fifth and sixth columns are the results with the maximum (50%) decaps. The sixth column lists the timing improvement rate over the delay increase due to power supply noise, which is defined as

$$\text{imp. rate} = \frac{\text{delay}_{w/o} - \text{delay}_w}{\text{delay}_{w/o} - \text{delay}_{Nom}}. \quad (11)$$

TABLE II
 DECAP ALLOCATION RESULTS COMPARED TO MAXIMUM DECAP ALLOCATION.

circuit	# cells	Nominal delay (ps)	Delay w/o decap (ps)	Max decap allocation		Proposed decap allocation		
				delay(ps)	imp. rate(%)	decap rate(%)	delay(ps)	imp. rate(%)
c432	232	716.1	906.0	902.3	1.96	25	896.3	5.07
c1355	329	399.7	495.7	497.2	-1.58	0	495.7	0
c1908	387	619.3	730.4	749.3	-17.1	0	730.4	0
c6288	3382	2371	2954	2947	1.29	25	2941	2.19
c7552	2070	608.9	751.5	758.7	-5.12	25	744.7	4.71
multiplier	41629	1590	1924	1930	-1.69	25	1914	3.15
ALU	14655	907.0	1103	1079	12.4	40	1077	13.3
average	-	-	-	-	-1.39	20	-	4.06

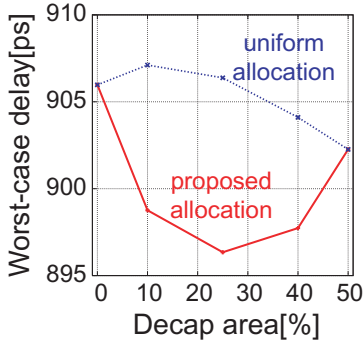


Fig. 9. Decap allocation results with various decap budget (c432).

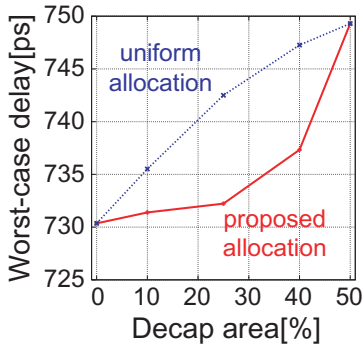


Fig. 10. Decap allocation results with various decap budget (c1908).

where $delay_{Nom}$ is the nominal delay without noise, $delay_{w/o}$ is the delay without decaps, and $delay_{w/}$ is the delay with decaps. The seventh to ninth columns are the results of the proposed method. The seventh column is the optimum amount of decaps to minimize the delay. Compared with the maximum allocation, the optimum allocation of decaps increases the improvement rate by 5.45% in average. In c1908 circuit, the improvement rate reaches 17.1%. Furthermore, the amount of decaps are reduced by 60% in average at the same time.

With the proposed method, we can effectively place decaps at timing-sensitive positions without increasing silicon area and gate leakage much.

We finally show an example of CPU time needed for the decap allocation. The CPU times for adjoint network analysis with a fast linear circuit simulator [10], the convolution in Eq. (9) and SSTA including criticality computation in Eq. (2) are 15.0s, 2.16s and 2.15s (e.g. c6288 circuit), respectively. The CPU time of the other computation is negligibly small.

V. CONCLUSION

In this paper, we proposed a timing-aware decap allocation method based on a statistical noise modeling. The proposed method considers that a large decap could lower supply voltage temporally, and explicitly computes the timing sensitivity to decap considering dynamic noise behavior and noise dependence on input patterns. The proposed method allocates decaps based on the sensitivities efficiently computed thanks to an approximation and adjoint sensitivity analysis. The experimental results show that the maximum decap allocation does not necessarily minimize the circuit delay. The proposed method gives better allocation compared to the maximum allocation and uniform allocation.

VI. ACKNOWLEDGEMENT

This work is supported in part by Semiconductor Technology Academic Research Center (STARC), New Energy and Industrial Technology Development Organization (NEDO) and VLSI Design and Education Center (VDEC).

REFERENCES

- [1] H. Li, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong, "Partitioning-Based Approach to Fast On-Chip Decap Budgeting and Minimization," in *Proc. DAC*, pp. 170–175, 2005.
- [2] H. Su, S. S. Sapatnekar, and S. R. Nassif, "Optimal Decoupling Capacitor Sizing and Placement for Standard-Cell Layout Designs," *IEEE Trans. CAD*, Vol. 22, No. 4, pp. 428–436, 2003.
- [3] S. Pant and D. Blaauw, "Timing-Aware Decoupling Capacitance Allocation in Power Distribution Networks," in *Proc. ASP-DAC*, pp. 757–762, 2007.
- [4] T. R.-Arabi, G. Taylor, M. Ma, and C. Webb, "Design & validation of the Pentium III and Pentium 4 processors power delivery," in *proc. VLSI Circuits*, pp. 220–223, 2002.
- [5] T. Enami, S. Ninomiya, M. Hashimoto, "Statistical Timing Analysis Considering Spatially and Temporally Correlated Dynamic Power Supply Noise," in *Proc. ISPD*, pp. 160–167, 2008.
- [6] OPENCORES.ORG, <http://www.opencores.org/>.
- [7] T. L. Pillage, R. A. Rohrer, and C. Visweswariah, "Electronic Circuit & System Simulation Methods," MC Graw-Hill, 1995.
- [8] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-Order Incremental Block-Based Statistical Timing Analysis," in *Proc. DAC*, pp. 331–336, 2004.
- [9] J. Xiong, V. Zolotov, N. Venkateswaran, and C. Visweswariah, "Criticality Computation in Parameterized Statistical Timing," in *Proc. DAC*, pp. 63–68, 2006.
- [10] C. Mizuta, J. Iwai, K. Machida, T. Kage, and H. Masuda, "Large-scale Linear Circuit Simulation with an Inversed Inductance Matrix," in *Proc. ASP-DAC*, pp. 511–516, 2004.