Optimal Decoupling Capacitor Sizing and Placement for Standard-Cell Layout Designs

Haihua Su, *Member, IEEE*, Sachin S. Sapatnekar, *Fellow, IEEE*, and Sani R. Nassif, *Senior Member, IEEE*

***Abstract*—With technology scaling, the trend for high-performance integrated circuits is toward ever higher operating frequency, lower power supply voltages, and higher power dissipation. This causes a dramatic increase in the currents being delivered through the on-chip power grid and is recognized in the 2001 International Technology Roadmap for Semiconductors as one of the difficult challenges. The addition of decoupling capacitances (decaps) is arguably the most powerful degree of freedom that a designer has for power-grid noise abatement and is becoming more important as technology scales. In this paper, we propose and demonstrate an algorithm for the automated placement and sizing of decaps in application specific integrated circuit (ASIC)-like circuits. The problem is formulated as one of nonlinear optimization and is solved using a sensitivity-based quadratic programming (QP) solver. The adjoint sensitivity method is applied to calculate the first-order sensitivities. We propose a fast convolution technique based on piecewise linear (PWL) compressions of the original and adjoint waveforms. Experimental results show that power grid noise can be significantly reduced after a judicious optimization of decap placement, with little change in the total chip area.**

***Index Terms*—Application specific integrated circuits (ASIC), decoupling capacitor, design automation, nonlinear programming, power distribution, sensitivity.**

I. Introduction

A. Motivation

MODERN designs are very sensitive to noise due to the lowering of supply voltages and the presence of a larger number of potential noise generators that eat significantly into the noise margins built into a design. The power grid, which provides the $V_{\rm dd}$ and ground signals throughout the chips, is one of the most important sources of noise, since supply voltage variations can lead not only to problems related to spurious transitions in some cases, particularly when dynamic logic is used, but also to delay variations [3] and timing unpredictability. Even if a reliable supply is provided at an input pin of a chip, it can deteriorate significantly within the chip due to the fact that the conductors that transmit these signals throughout the chip are electrically imperfect.

Manuscript received May 23, 2002; revised September 12, 2002. This work was supported in part by the National Science Foundation under Contract CCR-0098117 and in part by the Scientific Research Council under Grant 99-TJ-714. This paper was recommended by Guest Editor C. J. Alpert.

H. Su and S. R. Nassif are with the IBM Austin Research Laboratory, Austin, TX 78758 USA (e-mail: haihua@us.ibm.com; nassif@us.ibm.com).

S. S. Sapatnekar is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (sachin@ece.umn.edu).

Digital Object Identifier 10.1109/TCAD.2003.809658

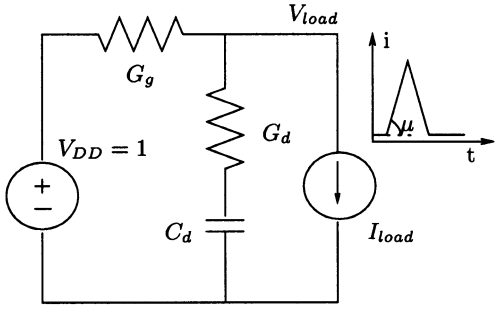

Fig. 1. Canonical and approximate circuit representation of a power network.

TABLE I
IC Technology Parameters

Year	L_{eff} nm	f MHz	V_{dd} V	Size mm^2	Power W	Density W/mm^2
2001	65	1684	1.1	310	130	*0.42*
2002	53	2317	1.0	310	140	*0.45*
2003	45	3088	1.0	310	150	*0.48*
2004	37	3990	1.0	310	160	*0.52*
2005	32	5173	0.9	310	170	*0.55*
2006	28	5631	0.9	310	180	*0.58*
2007	25	6739	0.7	310	190	*0.61*

A powerful technique for overcoming this problem is through the use of on-chip decoupling capacitors (decaps) that are intentionally attached to the power grid. To exemplify the role of decaps, let us consider the circuit shown in Fig. 1, which can be thought of as a canonical model of a power grid and loading circuit. In the figure, G_g models the grid conductance, G_d and C_d model a decoupling capacitance, and $I_{\rm load}$ models the time-dependent current waveform of the load, which we model for simplicity as

$$I_{\rm load} = \begin{cases} 0 & : t < 0 \\ \mu t & : t < t_p \\ \mu(2t_p - t) & : t < 2t_p \\ 0 & : t > 2t_p \end{cases} \tag{1}$$

where μ is the load current slope (unit: amps/s) and t_p (unit: s) is the time point when the current reaches its peak. We will use data from the 2001 International Technology Roadmap for Semiconductors [17], summarized in Table I, to predict the dependence of the load voltage $V_{\rm load}$ on the various circuit parameters in order to predict trends in power-grid-induced noise with technology scaling. The table shows the projected yearly trends for the effective length $L_{\rm eff}$, of a transistor, the circuit frequency, f, the supply voltage level, $V_{\rm dd}$, the chip size, the maximum

power dissipation, P, and the density of power dissipation per unit area, $P_{\square}$.

For the circuit shown in Fig. 1, we observe that $V_{\rm load}$ normalized by the voltage supply $V_{\rm dd}$ over the time interval from $t = 0$ to $t = t_p$ can be expressed as

$$V_{\rm load} = 1 - \frac{\mu}{G_g}\left(t - \frac{C_d}{G_g}(1 - e^{-t/\tau})\right) \tag{2}$$

where

$$\tau = \frac{(G_g + G_d)C_d}{G_g G_d}. \tag{3}$$

The minimum $V_{\rm load}$, or maximum normalized power-supply-induced noise occurs at $t = t_p$ and the magnitude of the noise is

$$V_{\rm max} = \frac{\mu}{G_g}\left(t_p - \frac{C_d}{G_g}(1 - e^{-t_p/\tau})\right). \tag{4}$$

We note that $t_p \propto f^{-1}$, and that power density (the last column in Table I, defined as power per unit area) $P_{\square} \propto V_{\rm dd}\mu t_p$, implying that $\mu \propto P_{\square} f/V_{\rm dd}$. Based on the trends in Table I, f increases by 4.0 $\times$ through the table, and μ increases by 9.13 $\times$. In order to keep $V_{\rm max}$ the same (i.e., keep the same amount of noise as a percentage of $V_{\rm dd}$), we need to dramatically increase the last term in (4): $C_d/G_g(1 - e^{-t_p/\tau})$. This means

- increasing the decoupling capacitance C_d, which can be done at the cost of small additional area, because the area efficiency of decoupling capacitance is expected to increase as the gate oxide is scaled,
- increasing the conductance associated with the decoupling capacitance G_d, which can be done by placing the capacitance *closer* to the load, and
- increasing the grid conductance G_g, which will be the most difficult to do because it goes somewhat against the prevailing scaling of interconnect, and the increased restrictions due to the consequent wire congestion emanating from this.

Unless we are able to do all of the above, it is likely that we will find the relative magnitude of power-grid-induced noise more than doubling by 2007.

The first two of these conclusions point convincing fingers toward the use of *appropriately placed* decaps for power grid noise reduction. While the use of decaps is certainly not new[1], the complexity of the problem requires shrewd optimal strategies driven by computer-aided design (CAD) tools, particularly in standard-cell environments in designs that require quick turnaround times in the face of strong time-to-market pressures.

Previous work [2], [6], [20] on decap allocation and optimization has focused on application in full custom design styles. A decap optimization procedure involving an iterative process of circuit simulation and floor planning is proposed in [6]. A linear programming technique is applied in [20] for allocation of white space for decap use and a heuristic is proposed to insert additional white space into an existing floorplan. Both [2] and [18] propose a sensitivity-based method of placing or optimizing decaps for reducing the noise, or voltage drop, in the power distribution network; the former method handles the problem in the frequency domain, and the latter in the time domain.

[1]For example, in a 300-MHz CMOS RISC Microprocessor design [5], as much as 160 nF of on-chip decoupling capacitance is added to control power-supply noise. In another example [10], the on-chip decoupling capacitance is sized at ten times that of the total active circuit switching capacitance.

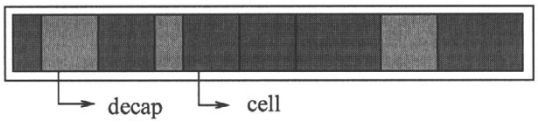

Fig. 2. One row of cells in a standard-cell layout showing decap locations.

B. Overview of Our Approach

In this work, we investigate the decap optimization and placement issue in the context of row-based standard-cell design typical of ASIC where each row has a fixed height. A reasonable hierarchical ASIC design flow develops designs for each functional block, which are then assembled at the chip level. It is important to ensure that the design of each such block incorporates the requirements of decap positioning for two reasons.

- The total area required by the decaps can be significant, and neglecting this can result in incorrect estimates of the block dimensions. An alternative could be to leave a certain percentage of the area of each block for decap insertion; however, this percentage is hard to arrive at, and the precise locations that should be left open for decap insertion is difficult to decide *a priori*.
- If the decision of decaps positioning is postponed until the entire layout is complete and the global power grid is designed, the amount of flexibility for decap positioning is limited. Consequently, the placed locations are likely to be suboptimal since decaps may have to be positioned far away from the points at which they are needed, which negates their strong ability to locally suppress power grid noise.

Therefore, we propose a design procedure for each functional block that uses a coarse global power grid model, described in Section II-A along with the internal power grid routing, and finds an optimal allocation of decaps to control the voltage drop in that block. Once these blocks have been designed and placed into the overall power grid, an upper-level power grid optimization or decap allocation technique can be applied to optimize the global power grid. Our work focuses on the former problem and does not address the latter.

For a standard-cell ASIC design, we consider a functional block inside a chip composed of N rows, with the ith row having M_i cells. Each of the N rows is filled by cells to some level of ratio $r_i (\leq 100\%)$. Decoupling capacitors can be placed in the empty space, which forms the $(1 - r_i)$ fraction of each row. One such row is illustrated in Fig. 2.

Our approach is designed to be applicable subsequent to the placement phase for the design of a functional block, where cells have already been assigned to rows. Since placement is designed to optimally place cells in order to achieve compactness for the layout and to control the wire length, timing and congestion, we use that result as the starting point for decap optimization, and perturb that solution in a minimal way in solving the *decap placement* problem. Because of this minimal pertur-

bation, the succeeding routing results are expected to be affected only slightly. Further more, timing driven placement will typically cram all the fastest cells together, which could potentially cause larger noise in power grid, a postplacement decap allocation to reduce noise becomes necessary.

Specifically, we propose to use the empty spaces that may be available within each row (when $r_i < 1$) to place decaps. In doing so, the exact position of each cell in that row is considered to be flexible although the order and the *relative* positions are fixed. Different placement of cells can lead to different widths and location of decaps, and consequently different impacts on the power supply noise, and the problem that we wish to tackle is that of finding the optimal cell placement which results in the minimization of a metric for the power supply noise. Note that since typical values of r_i are close to one, the major attributes of the original cell placement will be, for the most part, unaffected by our procedure.

The contributions of this work are as follows.

- We propose a nonlinear programming based decap optimization scheme for individual function blocks in standard-cell designs. The approach is performed after placement and has a minimal impact on the routing requirements.
- As a part of this procedure, we must calculate the sensitivities of a voltage drop metric using the adjoint network method. The direct application of this method results in very large amounts of data to be stored and convolved to calculate adjoint sensitivities, which leads to slow runtime as well as large memory usage. We develop an efficient and fast convolution technique based on piecewise linear (PWL) compressions of waveforms.

II. Power Supply Noise Metric and its Sensitivity Analysis

A. Modeling and Analysis

For the ASIC row-based standard-cell design style outlined above, it is common to use a predefined mesh-like power distribution network. As in [6] and [7], we model the network as follows.

- The power distribution network (grid) is abstracted as a resistive mesh.
- The cells are modeled as time-varying current sources connected between power and ground. Each current source waveform is obtained from other tools that determine the worst case input patterns. Various work on worst case current estimations can be found in [1], [12], [13], etc.
- The decoupling capacitors are modeled as single lumped capacitors connected between power and ground.
- The top-level metal is connected to a package modeled as an inductance connected to an ideal constant voltage source.

The behavior of such a circuit is described by a first order differential equation formulates using modified nodal analysis (MNA) [16]

$$Gx(t) + C\dot{x}(t) = u(t) \tag{5}$$

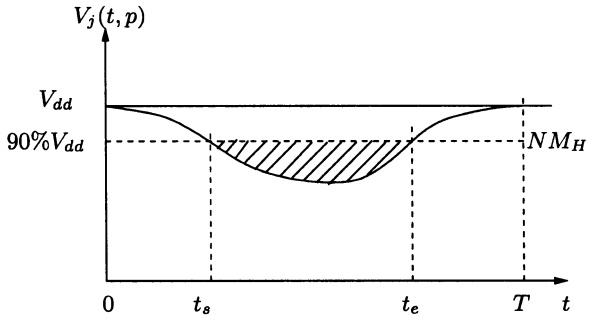

Fig. 3. Illustration of the voltage drop at a given node in the $V_{\rm dd}$ power grid. The area of the shaded region corresponds to the integral z at that node.

where x is a vector of node voltages and source and inductor currents; G is the conductance matrix; C includes both the decoupling capacitance and package inductance terms, and $u(t)$ includes the loads and voltage sources.

By applying the Backward Euler integration formula [16] to (5), we have

$$\left(G + \frac{C}{h}\right) x(t+h) = u(t+h) + x(t)\frac{C}{h} \tag{6}$$

where h is the time step for the transient analysis. If h is kept constant, only a single initial factorization of the matrix $G+C/h$ is required (as is done in [15] and [19]) leading to an efficient algorithm for transient analysis where each time step requires only a forward/backward solution step. After the transient analysis of the circuit, the voltage waveform at every node is known. Given that the treatment for nodes on the ground grid is completely symmetric, we restrict our discussion to the $V_{\rm dd}$ nodes for which we formally define the drop at node n to be simply $V_{\rm dd} - V_n(t)$, where $V_n(.)$ signifies the voltage at node n.

An efficient metric to estimate power-grid-induced noise at a node is the integral of the voltage drop below a user specified noise ceiling [8]

$$\begin{aligned} z_j(p) &= \int_0^T \max\{NM_H - v_j(t,p), 0\}dt \\ &= \int_{t_s}^{t_e} \{NM_H - v_j(t,p)\}dt \end{aligned} \tag{7}$$

where p represents the tunable circuit parameters which, in our case, are the *widths* of the decoupling capacitors[2]. The voltage drop integral beyond the expressed by (7) represents the shaded area in Fig. 3. We define the measure of goodness for the whole circuit as the sum of the individual node metrics

$$Z = \sum_{j=1}^{K} z_j(p) \tag{8}$$

where K is the number of nodes. This metric penalizes more harshly transients that exceed the imposed noise ceiling by a large amount for a long time, and has empirically been seen to be more effective in practice than one that penalizes merely the maximum noise violation. Intuitively, this can be explained by the fact that the metric incorporates, in a sense, both the voltage

[2]We choose the width since the height of the decoupling capacitors is constrained to be the same as the height of the functional cells in the same row, as illustrated in Fig. 2.

and time axes together, as well as spatial considerations through the summation over all nodes in the circuit.

B. Integral Sensitivity Computation

Adjoint sensitivity analysis is a standard technique for circuit optimization where the sensitivity of one performance function with respect to many parameter values is required [9], [11], [16]. For our problem, the use of this method is a natural choice since we are interested in the sensitivity of the scalar objective function (8) with respect to the widths of all decaps in the network.

An adjoint network with the same topology as the original network is constructed, with all of the voltage sources in the original network shorted and current sources open. For noise functions of the form given in (7), the adjoint network will include a current source of value $-u(t-t_s)+u(t-t_e)$ applied at node j if $z_j \neq 0$. We set the initial conditions to the adjoint circuit to zero and analyze it backward in time. We use the same time step h as the original circuit, thus allowing us to reuse the previously computed LU factorization for $(G+C/h)^{-1}$. Consequently, the extra simulation cost is reduced to one forward/backward solve for each time step of the adjoint circuit. Obviously, a smaller timestep results in a higher accuracy for both the original and adjoint waveforms, and consequently higher accuracy in the sensitivities at the expense of a longer runtime. We find that in order to ensure the accuracy of adjoint sensitivities, using 500–1000 steps per clock cycle (i.e., $h = 0.002T_{\text{period}}$ or $0.001T_{\text{period}}$) is sufficient.

The sensitivity of the objective function with respect to all of the decoupling capacitors in the circuit can be computed from the following convolution [9], [11]:

$$\frac{\partial Z}{\partial C} = \int_0^T \psi_C(T-t)\dot{v}_C(t)dt \tag{9}$$

where $\psi_C(\tau)$ is the waveform across the capacitor C in the adjoint circuit.

C. Improving the Efficiency of Adjoint Sensitivity Calculation

In our context, we cannot use the above adjoint sensitivity approach directly, and must tailor it to control the storage required by the direct application of this method and speed-up the convolution calculation shown in (9). Specifically, a significant complication arises in the case of very large networks where the total amount of data to be stored is proportional to the number of nodes multiplied by the number of time steps, and could reach 10^9 bytes or more for large networks with millions of nodes[3]. In order to alleviate the problem, we store the waveforms of the original and adjoint network using a compressed PWL form. This results in a situation of the type illustrated in Fig. 4, where the time points on the original and adjoint waveforms are not aligned. However, since we know that waveforms are divided by linear segments, the convolution (9) of the waveforms $\psi_C(\tau) = g+k\tau$ and $v_C(t) = a+bt$ over the time interval $[p, q]$ can be expressed as

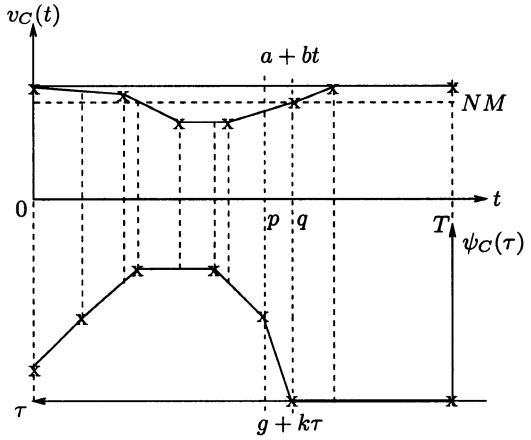

Fig. 4. Compressed PWL waveforms for the original and adjoint networks.

$$\begin{aligned}\int_p^q \big(g+k(T-t)\big)\,\frac{d(a+bt)}{dt}\,dt &= \int_p^q \big(g+k(T-t)\big)\,b\,dt \\ &= b(q-p)\left(g-kT-k\left[\frac{q-p}{2}\right]\right).\end{aligned} \tag{10}$$

The complexity of the convolution calculation over $[0, T]$ is $O(N+M)$, where N and M are the number of linear segments on the original and adjoint waveforms.

Once the sensitivities of Z with respect to all of the decoupling capacitor values are computed, the sensitivities to the *width* of each capacitor can be calculated using the chain rule, as in [18]

$$\frac{\partial Z}{\partial w} = \frac{\partial Z}{\partial C} \times \frac{\partial C}{\partial w}. \tag{11}$$

Given that we calculate the decoupling capacitance from

$$C = \frac{\varepsilon_{ox}}{T_{ox}} \times w \times h \tag{12}$$

where T_{ox} and ε_{ox} are the thickness and permittivity of the gate oxide, and h is the fixed height of the decap, it is easily verified that (11) becomes

$$\frac{\partial Z}{\partial w} = \frac{\partial Z}{\partial C} \times \frac{\varepsilon_{ox}}{T_{ox}} \times h \tag{13}$$

III. Optimization and Placement

A. Problem Formulation

The problem of decoupling capacitor optimization is now formulated as

$$\begin{aligned}&\textbf{Minimize } Z(w_j) \;\; j=1\ldots N_{\text{decap}} \\ \textbf{Subject to } &\sum_{k\in row_i} w_k \leq (1-r_i)W_{\text{chip}} \;\; i=1\ldots N_{\text{row}} \\ &\textbf{and } 0 \leq w_j \leq w_{\max} \;\; j=1\ldots N_{\text{decap}}.\end{aligned}$$

The scalar objective Z, defined in (8), is a function of all of the decap widths and N_{decap} is the total number of decaps in the

[3]Despite our total number of nodes being in the order of thousands and memory not being an issue, the speed-up of adjoint sensitivity calculation shown in our experiments is significant with very small accuracy tradeoff.

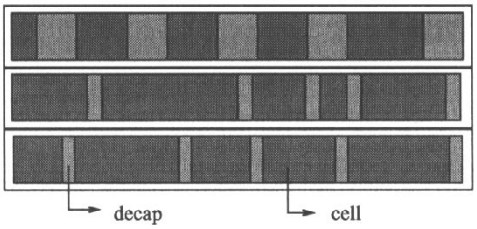

Fig. 5. Illustration of the initial equal distribution of decaps.

chip. The first constraint states that the total decap width in a row cannot exceed the total amount of empty space in that row, and W_{chip} and N_{row} denote, respectively, the width of the chip and the number of rows in the chip. The second constraint restricts the decap widths within a realistic range. An upper bound $w_{\max}$ for a cell in row i is easily seen to be $(1 - r_i)W_{\text{chip}}$, which is the largest empty space in row i, while the lower bound of each decap width is zero.

Equation (14) represents a linearly constrained nonlinear optimization problem. The objective function Z can be obtained after the transient analysis of the power grid circuit, and its sensitivity with respect to all of the variables w_j can be calculated using the adjoint method discussed in Section II-B. We choose to use a standard QP solver [21] for solving large nonlinear optimization problems. We start the optimization with an initial guess that uniformly distributes the vacant space in each row to each decoupling capacitor in each row, as illustrated in Fig. 5. It can be seen that initially there is one decap next to each cell. The initial chip width is chosen to be the maximum width occupied by cells and decaps among all rows.

Since the QP solver [21] solves large unconstrained nonlinear optimization problems with simple bounds (lower and upper bounds on the variables), we apply the Lagrangian relaxation technique [4] which adds constraint functions into the objective function. For each row i, the nonnegative relaxation variable S_i is chosen such that

$$\phi_i = \sum_{k \in \text{row}_i} w_k - (1 - r_i)W_{\text{chip}} + S_i^2 = 0, \; i = 1 \cdots N_{\text{row}}, \tag{14}$$

where S_i is bounded within a small range (say, $W_{\text{chip}}/10$) which allows little change of total decap area within every row i. By further introducing Lagrangian variables λ_i for each row, the new objective function becomes

$$f = Z + \sum_i \lambda_i \phi_i \tag{15}$$

where λ_i is unbounded ([21] can handle unbounded variables). Theoretically, function f has the same minimum as the original objective function Z [14] and the new problem size is $N_{\text{decap}} + 2N_{\text{row}}$.

B. Optimization and Placement Scheme

The optimization procedure invokes the QP optimizer, and the set of steps that are repeated during each iteration of the optimizer can be summarized as follows.

1) Perform the transient simulation of the original power grid circuit and store PWL waveforms of all decaps.
2) Check all nodal voltages for those that fall below the noise margin, identify hot spots and compute the objective function Z.
3) Set up the sources corresponding to these failure nodes for the adjoint circuit.
4) Perform the transient simulation of the adjoint circuit and store PWL waveforms of all decaps.
5) Compute the sensitivities $\partial Z/\partial C_j$ by convolution and use the chain rule to obtain $\partial Z/\partial w_j$.
6) Compute the constraint function and its Jacobian.
7) Feed all of the information into a QP solver and update the vector of widths $\vec{w}$ according to the values returned by the solver.
8) According to the updated $\vec{w}$, reposition all of the cells and decaps in the row from left to right.

C. Extensions

With only slight changes to the original problem formulation, our method can be extended to handle: 1) a special case with cell alignment restrictions and 2) a general nonstandard-cell placement case.

1) Cell Placement With Vertical Alignment Restrictions: Row-based placement often gives vertical alignment for critical cells that require either sets of cells to be aligned in terms of their left edges or right edges. We denote the lower coordinate of a cell or a decap by X_{low} and its higher coordinate by X_{high}. Since X_{low} [X_{high}] of a cell is the same as X_{high} [X_{low}] of the decap to its left [right], so that it is sufficient to use the decap coordinates only.

Given two cells i and j, the vertical alignment restriction on their x coordinates can be directly translated to the restrictions on the two decaps adjacent to them. Assume S_1 and S_2 are the sets of decaps with alignment restrictions on X_{low} and X_{high}, respectively. By changing the decision variables in the previous formulation to the lower and higher coordinates of the decaps, the problem of optimization with vertical alignment cells can be stated as a constrained NLP as follows:

$$\begin{aligned}
&\textbf{Minimize } Z(X_{\text{low},j}, X_{\text{high},j}) \;\; j = 1 \ldots N_{\text{decap}} \\
&\textbf{Subject to } \sum_{k \in row_i} (X_{\text{high},k} - X_{\text{low},k}) \\
&\qquad \le (1 - r_i)W_{\text{chip}} \;\; i = 1 \ldots N_{\text{row}} \\
&\qquad X_{\text{high},m} = X_{\text{high},n} \;\; m, n \in S_1 \\
&\qquad X_{\text{low},p} = X_{\text{low},q} \;\; p, q \in S_2 \\
&0 \le X_{\text{high},j} - X_{\text{low},j} \le w_{\max} \;\; j = 1 \ldots N_{\text{decap}} \\
&\textbf{and } 0 \le X_{\text{high},j}, X_{\text{low},j} \le W_{\text{chip}} \;\; j = 1 \ldots N_{\text{decap}}.
\end{aligned}$$

2) Nonstandard-Cell Placement: Given a general layout, as shown in Fig. 6, a simple extension of our algorithm is to heuristically divide the problem into two NLP problems, one optimizing in the vertical direction only and the other in the horizontal direction. The decision variables for the NLP in the vertical direction are the heights of each decap, while the decision variables for the horizontal problem are the widths of each decap.

The overall scheme can be iteratively performed as follows.

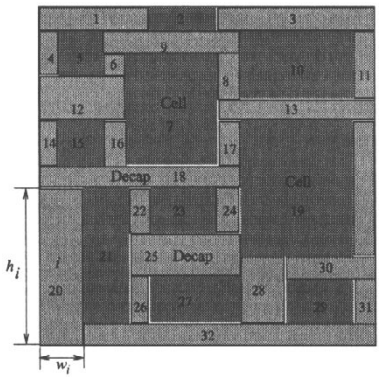

Fig. 6. General placement.

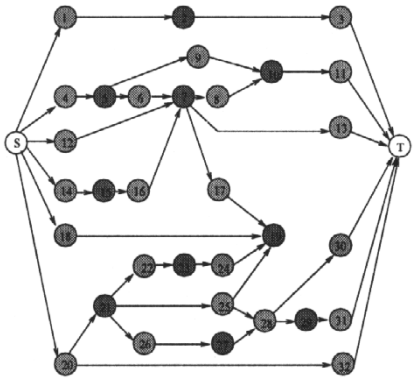

Fig. 7. Horizontal constraint graph of the placement shown in Fig. 6.

1) Construct the horizontal and vertical constraint graph according to the given initial placement, as in [6] and [20]. In a constraint graph, a node represents a decap or a cell. The weight of each node in the horizontal/vertical constraint graph is assigned to be the width/height of the cell or decap. The horizontal constraint graph of Fig. 6 is shown in Fig. 7.
2) Derive constraints on the amount of empty space in the X direction from every path in the horizontal constraint graph. For example, in Fig. 7, the constraint corresponding to path $(S \rightarrow 4 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow T)$ is $w_4 + w_9 + w_{11} <= W_{\text{chip}} - (w_5 + w_{10})$, where W_{chip} is the chip width and w_5 and w_{10} are constants. Using QP solver, solve the horizontal constrained NLP problem with respect to decap widths, $\vec{w}$.
3) According to the updated $\vec{w}$, reposition all of the cells and decaps.
4) Derive constraints on the amount of empty space in the Y direction from every path in the vertical constraint graph. Using QP solver, solve the vertical constrained NLP problem with respect to decap heights, $\vec{h}$.
5) According to the updated $\vec{h}$, reposition all of the cells and decaps.

TABLE II
WAVEFORM COMPRESSION RESULTS

Blk	Num of decaps	ϵ (V)	memory (MB)	CPU time (sec)	avg sens err%	Z^* err%
1	1964	0.0	60.7	0.66	0	0
		10^{-6}	14.3	0.17	0.37	0.00118
2	3288	0.0	85.9	1.10	0	0
		10^{-6}	19.8	0.30	4.6e-3	0.00007
3	3664	0.0	93.0	1.25	0	0
		10^{-6}	21.0	0.36	0.20	0.00023

*Z is sum of noise integrals.

TABLE III
PERFORMANCE TRADEOFF VERSUS THE VALUE OF ϵ

ϵ (V)	memory (MB)	CPU time (sec)	avg sens err%	Z^* err%
0.0	60.7	0.66	0	0
10^{-6}	14.3	0.17	0.37	0.0012
10^{-5}	13.9	0.16	1.12	0.0503
10^{-4}	13.4	0.14	2.66	1.1390
10^{-3}	13.2	0.13	3.14	12.385
10^{-2}	13.1	0.13	10.4	16.301

*Z is sum of noise integrals.

The iteration stops when no more improvement for the power grid noise can be achieved.

IV. EXPERIMENTAL RESULTS

The proposed decap optimization and placement scheme has been integrated into a linear circuit simulator written in C++ and the QP solver is applied. All experimental results are performed on a 1.8-GHz Pentium IV machine under the Redhat Linux operating system. We work on three functional blocks in an industrial ASIC design, which are referred to as Blocks 1, 2, and 3. Each of them is a 0.18-μm CMOS design operating under a supply voltage of 1.8 V.

We first look at the performance of our PWL waveform compression technique in Table II. For each functional block, the total number of decaps are listed in column 2. In column 3, ϵ is defined as an upper bound for the voltage difference between the actual simulated value and the one approximated by the PWL equation. When the difference exceeds $\epsilon(V)$, one breakpoint of the waveform is stored, otherwise, the point is removed. When ϵ is zero, the waveform at every timestep is stored and the sensitivity result is the most accurate. Columns 4 and 5 show the total memory and CPU time used during the waveforms convolution. Column 6 shows the average percentage error of the calculated sensitivities with respect to the accurate values among all decaps in the block. The last column shows the percentage error of sum of noise integrals, Z, which is the objective function of our optimization problem. It can be seen that the memory and CPU time reduction are each around 4 $\times$ in all cases, the loss of accuracy in sensitivity is within 0.4% by average and the loss of accuracy in Z is within 0.002%, which is negligible.

Table III shows the performance tradeoff for various ϵ values for Block 1. The data show a slightly greater memory and CPU time reduction as ϵ increases, while the average percentage error of sensitivity goes to around 10% and the percentage error of Z goes to around 16%. In the following experiments, we choose ϵ as $10^{-6}(V)$.

TABLE IV
OPTIMIZATION RESULTS

Block		Num bad nodes	Num of nodes	ΔV_{max} (V)	Z^* $(V\times ns)$	Num of rows	Num of decaps	Problem size	Num of valid decaps	CPU time (min)
1	Before	105	974	0.193	0.121	53	1964	2070	1964	0.90
	After	0		0.176	0.000					
2	Before	80	861	0.230	0.366	85	3288	3458	2240	15.2
	After	63		0.196	0.063					
3	Before	100	828	0.222	0.649	132	3664	3928	3430	12.5
	After	70		0.201	0.200					

Before = Before optimization; After = After optimization
*Z is sum of noise integrals.

Table IV lists the decap optimization results for the three functional blocks. The occupancy ratio r_i for each row of these blocks is around 80%. Initially, decaps are uniformly distributed across each row between each cell, so that the number of cells is roughly equal to the number of decaps. The results in the table before optimization, therefore, correspond to this uniform distribution of decaps. In Table IV, the second column shows the number of nodes with noise violations (i.e., nodes j with a nonzero value of z_j) before and after optimization; the total number of nodes in the power grid are shown in the third column. Although the power grid size of each block is not large, as discussed in Section I-B, we emphasize that our problem addresses a hierarchical design style in which the whole chip is divided into smaller functional blocks, and the decap optimization of each block is performed individually to fully exploit the localized nature of the noise suppression effect of decaps. The next two columns compare the worst case voltage drop and the sum of integral area Z (i.e., the original objective function) before and after optimization. Of the three examples, the worst case (Block 3) noise (Z) reduction is about one-third of the initial value, which corresponds to the uniform distribution of decaps. The significant change in the value of Z before and after optimization further supports our earlier claim of the strong local effects of the decaps and the need for a hierarchical design methodology in which decaps are inserted into functional blocks during design rather than as an afterthought. Column 6 shows the total number of rows in the block. The total number of decoupling capacitors placed in the whole block is listed in column 7. Column 8 shows the problem size ($N_{\rm decap} + 2N_{\rm row}$) for the Lagrangian form discussed in Section III–A. We set the lower bound of each decap as zero for the optimization because one can imagine that near some cells, the voltage drop is so small that no decap is required. The decap widths returned by the optimizer are continuous between zero and the upper bound. The actual manufactured decaps are restricted to the smallest transistor size, which, in our experiments, is assumed to be 0.36 μm (2λ). We define a valid decap as one whose width is larger than 0.36 μm. Total number of valid decaps after optimization are listed in column 9 of the table. We have verified that after the removal of all of the tiny decaps (i.e., those whose widths are less than a threshold), the total power grid noise Z and the maximum voltage drop of each circuit remain unchanged. Finally, the last column lists the total amount of CPU time to run each example. For each of these three blocks, the worst case voltage drops and sum of the integral areas are both reduced successfully.

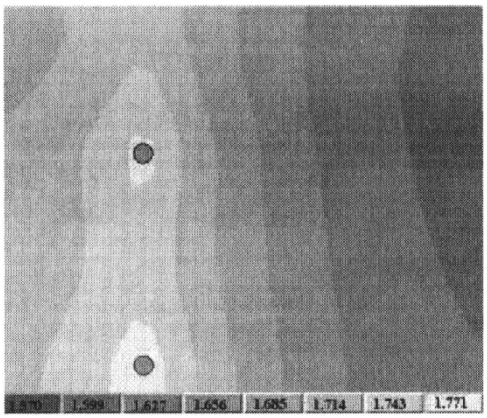

Fig. 8. Voltage drop contour on the $V_{\rm dd}$ plane before optimization.

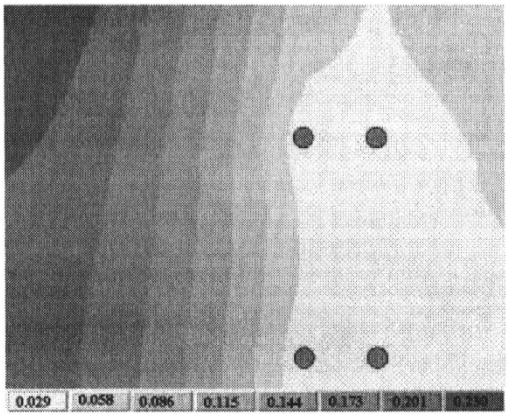

Fig. 9. Voltage drop contour on the ground plane before optimization.

The $V_{\rm dd}$ and ground contour of Block 2 are shown in Figs. 8 and 9. The small ovals in each figure represent VDD or GND c4 locations. In both figures, each gray-scaled color corresponds to a voltage drop range and the number written in each color sample shows the lowest voltage drop in that range. Darker colors mean larger voltage drops. It can be seen that the voltage range in the $V_{\rm dd}$ plane is 1.610–1.8 *V* and the hot spot is located on the right side of the block. Similarly, the voltage range in the ground plane is 0–0.230 *V*, and the hot spot is located on the left side of the block. The result of the optimal cell and decap placement for Block 2 is shown in Fig. 10. We observe that this placement is consistent with the hot spots of the block, i.e., larger decaps are allocated closer to the two sides of the block. After optimization, the voltage drop in the $V_{\rm dd}$ plane is in the range of 0–0.196 *V* and that of the ground plane is in the

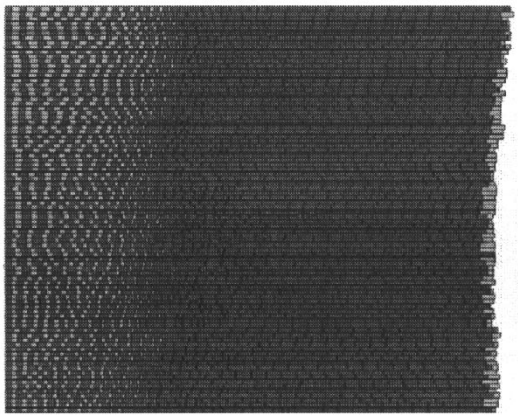

Fig. 10. Results of the decap placement algorithm on Block 2. The dark regions represent the standard cells and the light regions represent the decaps.

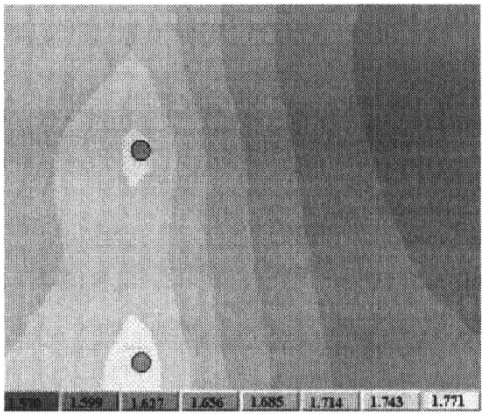

Fig. 11. Voltage drop contour on the $V_{\rm dd}$ plane after optimization.

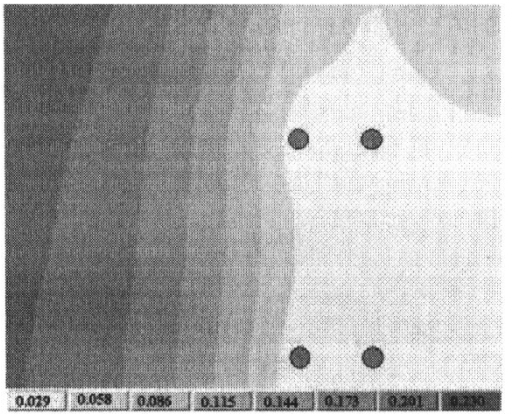

Fig. 12. Voltage drop contour on the ground plane after optimization.

range of 0–0.191 *V*. The optimization process has judiciously balanced the power grid voltage drop on the whole block. For comparison, Figs. 11 and 12 show the voltage contour for each plane after optimization. It should be noted that decap placement is not the only method for noise reduction, and that other techniques such as wire widening, or increasing the density of the power grid, can be applied to further improve the power grid performance. Therefore, these results that holistically reduce the degree of noise violation by decap placement correspond to a first step in power grid optimization, and can be supplemented by other techniques to obtain a solution that satisfies the noise constraints imposed on the design as the global grid is designed.

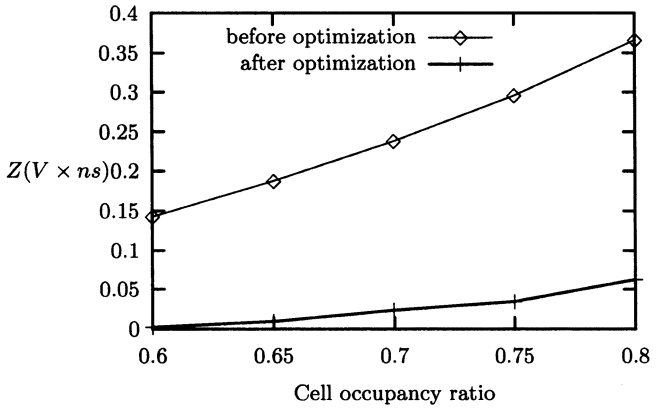

Fig. 13. Variation of the noise metric with the occupancy ratio (Block 2).

The noise reduction trend with respect to the cell occupancy ratio r_i for Block 2 is shown graphically in Fig. 13. This experiment is performed by removing some cells from each row of the block to achieve the desired occupancy ratio. For each case, around 10% of the total grid nodes are beyond the noise margin. A block with lower occupancy ratios provides more empty space for decoupling capacitors and, consequently, is easier to optimize. Therefore, in Fig. 13, the noise reduction is more efficient for cases with lower occupancy ratios than for those with higher ones.

Our decap optimization slightly perturbs the original timing-driven placement, and, therefore, it is necessary to see how much the routing performance can be affected. To test this, we performed global routing for each block before and after optimization. In the global router, the entire block region is divided into small tiles and the wire density on a tile boundary is defined as the ratio between total number of wires across the boundary and its wiring capacity. In Table V, the total number of cells and nets are listed in columns 3 and 4. The block size and total number of tiles used for global routing are provided in columns 5 and 6. After global routing, the total wire length (in terms of Manhattan distance) and maximum wire density among all tile boundaries are shown in columns 7 and 9. As can be seen from the percentage change in the total wire length (column 8) and the change of maximum wire density (column 10) in Table V, the routing performance is only slightly affected and is not always worsened. The experiments in Table V correspond to a maximum decap occupancy ratio of 80%, and perturbing placements to allow larger decap occupancy ratios could cause larger changes in the routing results.

V. Conclusion

This paper has presented an on-chip decoupling capacitor sizing and placement scheme aimed at making the best use of empty spaces in the row-based standard-cell design of ASICs. The problem of decap insertion and placement has been motivated for current and future technologies, and the problem has been formulated as a constrained nonlinear optimization problem that is successfully solved using the gradient-based QP

TABLE V
ROUTING PERFORMANCE BEFORE AND AFTER DECAP OPTIMIZATION

Block		Num of cells	Num of nets	Block size ($\mu m \times \mu m$)	Tile num	Total wire length(μm)	Change% of wire length	Max wire density	Change% of max wire density
1	Before	1964	2553	818x648	18x34	3009338		0.991	
	After					3011272	+0.06	1.000	+0.91
2	Before	3288	5255	1340x1048	29x55	9165767		0.842	
	After					8944180	-2.42	0.847	+0.59
3	Before	3664	7256	795x1644	46x33	12992807		0.984	
	After					13166525	+1.34	0.996	+1.22

solver. For a predesigned power distribution network, the location and size of each decap is updated iteratively such that the total transient noise in the power grid is minimized, and the technique is demonstrated on several industrial designs.

REFERENCES

[1] G. Bai, S. Bobba, and I. N. Hajj, "Emerging power management tools for processor design," in *Proc. Int. Symp. Low Power Electron. Design*, Monterey, CA, Aug. 1998, pp. 143–148.

[2] ——, "Simulation and optimization of the power distribution network in VLSI circuits," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2000, pp. 481–486.

[3] G. Bai, S. Bobba, and T. N. Hajj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *Proc. Design Automation Conf.*, Las Vegas, NV, June 2001, pp. 295–300.

[4] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York: Wiley, 1993.

[5] W. J. Bowhill, R. L. Allmon, and S. L. Bell, "A 300 MHz 64b quad-issue CMOS RISC microprocessor," in *Proc. Int. Solid-State Circuits Conf.*, Piscataway, NJ, Feb. 1995, pp. 182–183.

[6] H. H. Chen and D. D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," in *Proc. Design Automation Conf.*, Anaheim, CA, June 1997, pp. 638–643.

[7] H. H. Chen and J. S. Neely, "Interconnect and circuit modeling techniques for full-chip power supply noise analysis," *IEEE Trans. Comp. Packag. Technol.*, pt. B, vol. 21, pp. 209–215, Aug. 1998.

[8] A. R. Conn, R. A. Haring, and C. Visweswariah, "Noise considerations in circuit optimization," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 1998, pp. 220–227.

[9] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, vol. 16, pp. 318–323, Aug. 1969.

[10] D. W. Dobberpuhl, R. T. Witek, and R. Allmon, "A 200-MHz 64-b dual-issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1555–1567, Nov. 1992.

[11] P. Feldmann, T. V. Nguyen, S. W. Director, and R. A. Rohrer, "Sensitivity computation in piecewise approximate circuit simulation," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 171–183, Feb. 1991.

[12] Y.-M. Jiang, K.-T. Cheng, and A.-C. Deng, "Estimation of maximum power supply noise for deep sub-micron designs," in *Proc. Int. Symp. Low Power Electronics Design*, Monterey, CA, Aug. 1998, pp. 233–238.

[13] H. Kriplani, F. Najm, and I. Hajj, "Maximum current estimation in CMOS circuits," in *Proc. Design Automation Conf.*, Anaheim, CA, June 1992, pp. 2–7.

[14] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.

[15] S. R. Nassif and J. N. Kozhaya, "Fast power grid simulation," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2000, pp. 156–161.

[16] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic and System Simulation Methods*. New York: McGraw–Hill, 1995.

[17] *The International Technology Roadmap for Semiconductors* , Semiconductor Ind. Assoc., 2001.

[18] H. Su, K. H. Gala, and S. S. Sapatnekar, "Fast analysis and optimization of power/ground networks," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2000, pp. 477–480.

[19] M. Zhao, R. V. Panda, S. S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, "Hierarchical analysis of power distribution networks," in *Proc. Design Automation Conf.*, Los Angeles, CA, June 2000, pp. 481–486.

[20] S. Zhao, K. Roy, and C.-K Koh, "Decoupling capacitance allocation for power supply noise suppression," in *Proc. Int. Symp. Physical Design*, Napa, CA, Apr. 2001, pp. 66–71.

[21] C. Zhu, R. H. Byrd, and J. Nocedal, *LBFGS-B: Fortran Subroutines for Large-Scale Bound Constrained Optimization*. Evanston, IL: Northwestern Univ., 1994.

Haihua Su (S'00–M'03) received the B.S. degree from Hefei University of Technology, Hefei, China, in 1995, the M.S. degree from the University of Science and Technology of China, Hefei, China, in 1998, and the Ph.D. degree from the University of Minnesota, Minneapolis, in 2002, all in electrical engineering.

She is currently a Postdoctorate Fellow with the IBM Austin Research Laboratory, Austin, TX. She has been working on VLSI CAD issues related to global interconnects, in specific, power and clock network analysis and optimization, codesign of power, and signal networks.

Sachin S. Sapatnekar (S'86–M'93–SM'99–F'03) received the B.Tech. degree from the Indian Institute of Technology, Bombay, India, in 1987, the M.S. degree from Syracuse University, Syracuse, NY, in 1989, and the Ph.D. degree from the University of Illinois, Urbana–Champaign, in 1992.

From 1992 to 1997, he was an Assistant Professor in the Department of Electrical and Computer Engineering, Iowa State University. He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. He has coauthored two books and coedited one book in the areas of timing and layout optimization.

Prof. Sapatnekar has been an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING, has served on the Technical Program Committee for various conferences, as Technical Program and General Chair for the Tau workshop and the International Symposium on Physical Design. He is currently a Distinguished Visitor for the IEEE Computer Society and a Distinguished Lecturer for the IEEE Circuits and Systems Society. He is a recipient of the National Science Foundation Career Award and best paper awards at the 1997 and 2001 Design Automation Conferences and the 1998 International Conference on Computer Design.

Sani R. Nassif (S'80–M'86–SM'02) received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1986.

He worked for ten years at Bell Laboratories on various aspects of design and technology coupling including device modeling, parameter extraction, worst case analysis, design optimization, and circuit simulation. He joined the IBM Austin Research Laboratory, Austin, TX, in 1996, where he is presently managing the Tools and Technology Department, which focuses on design/technology coupling, timing simulation and analysis, testing, and low power design.