Optimal Placement of Power Supply Pads and Pins

Min Zhao, Yuhong Fu, Vladimir Zolotov, Savithri Sundareswaran, and Rajendran Panda
Motorola, Inc., Austin, TX

ABSTRACT

Power delivery networks of VLSI chips require adequate input supply connections to ensure reliable performance. This paper addresses the problem of finding an optimum set of pads, pins, and on-chip voltage regulators, and their placement in a given power supply network, subject to constraints on the voltage drops in the network and maximum currents through the pads, pins and regulators. The problem is modeled as a mixed integer linear program using macromodeling techniques and several heuristic techniques are proposed to make the problem tractable. The effectiveness of the proposed techniques is demonstrated on several real chips and memories used in low-power and high-performance applications.

Categories and Subject Descriptors: B.7.2
General Terms: Algorithms, Design, Performance
Keywords: Pad optimization, pad placement

1. INTRODUCTION

With the increase in complexity of VLSI circuits and supply voltage scaling, designing a power distribution network has become a challenging task. A robust power network design is essential to ensure that circuits on a chip operate reliably at the guaranteed level of performance. A poorly designed power network can cause a variety of problems such as loss of circuit performance, noise generation, and electro-migration failures. In view of this, much research has been directed recently towards several areas of power network design. [1] presented efficient power grid simulation techniques. [2] proposed a macromodeling technique for solving multi-million size power grids very efficiently. Wire sizing and topology optimization for power distribution networks were addressed in [3, 4, 5, 6, 7].

In this paper, we address another aspect of power network design, viz. finding an optimal set of power pads, pins, or voltage regulators, and their placement in the design such that on-chip voltages are kept within a target level and currents through them are limited to a specified maximum. Given a fully placed and power-routed design, we propose a method for determining a minimum number of locations at which the supply connections should be made.

This method can be used in two design scenarios:

- at the chip level, to determine the minimum number of power and ground pads and/or voltage regulators and their locations.
- at the SoC (System-on-Chip) block level, to determine the recommended minimum number and locations of power and ground pins which must be serviced by the chip-level power/ground routes.

The number of supply pads, pins, or voltage regulators (hereafter referred collectively as *pads* for convenience) required for a chip, core, or memory can be very few (less than 10) or very large (few thousands), depending on its power consumption and the size and style of power network design. In all cases, the task of finding an optimal set of locations for them is quite difficult due to a potentially large set of candidate locations from which to choose. For a chip which is designed for wire-bond package, for example, the candidate locations are typically all possible pad locations on the peripheral power ring. In case of a high performance processor using a flip-chip package, a matrix of locations determined by the pitch of the C4 ball array forms the candidate set. In case of SoC blocks such as memory, core, or custom macro, the candidate set is much larger, consisting typically of all terminations of power stripes reaching up to the periphery of the block. If over-the-block power routing is planned, then all interior points on the upper power/ground layers of the design that can be reached by the routes over the block need to be considered as candidates.

The proposed optimization targets to guarantee the following under a specified set of varying load conditions:

- Voltages at devices (transistors and cells) are better than a set target. This is to ensure correct circuit operation at expected level of performance.
- Current supplied by a pad, pin, or regulator is within a specified limit. This is required (i) for not exceeding the design capacity of regulators and pads and (ii) to distribute currents more uniformly among the pads so that the $L\ di/dt$ voltage variations due to parasitic inductance in the package's substrate, ball grid array and bond wires are minimized.

The only previous work, to the best of our knowledge, relating to power pad optimization is [8]. It gives a heuristic for simultaneous pad assignment and power routing. While [8] tries to lower the di/dt noise using minimum routing and considers only multi-tree topologies, we reduce both IR drop and maximum pad current, the latter for reducing Ldi/dt noise. Moreover, our method is applicable to any power grid topology, tree or mesh.

Today's VLSI power grids are extremely large (hundreds of million nodes in size) and it takes several minutes to solve them even for one static (DC) load condition. Needless to say, exploring all possible pad assignments (which is exponential on the number of candidate pad locations) is intractable. We formulate this task as a Mixed Integer Linear Programming (MILP) optimization, using previously proposed hierarchical macromodeling techniques[2]. It is formulated such that the

DAC 2004, June 7–11, 2004, San Diego, California, USA.

number of variables and constraints are linear on the number of pad candidate locations and a small set of nodes, called the *observation nodes* (to be explained later). MILP is very expensive for large number of integer variables, which unfortunately equals the number of pad candidates in our formulation. But, for a high quality solution, it is desirable to consider a large candidate set. Therefore, we need some efficient heuristics to reduce the run-time complexity of solving the problem with a large candidate set. In addition to the traditional branch-and-bound heuristic, we employ two other heuristics: (i) Pruning of pad candidate set and (ii) Divide-and-conquer.

We will first describe the method to optimize for one static (DC) current demand pattern in the design, and later provide an extension to consider multiple DC loading scenarios. Thus, the proposed solution will guarantee target voltage levels for a variety of *specified* load patterns.

2. PRELIMINARIES

An RLC power network with independent time-varying current sources representing the switching currents of the transistors and gates, and Norton current sources for supply connections, can be simulated in a typical Modified Nodal Analysis approach:

$$\mathbf{G} \cdot \mathbf{x}(t) + \mathbf{C} \cdot \mathbf{x}'(t) = \mathbf{b}(t), \tag{1}$$

where $\mathbf{G}$ is a conductance matrix, $\mathbf{C}$ is a admittance matrix resulting from capacitive and inductive elements, $\mathbf{x}$(t) is a time-varying vector of nodal voltages and inductor currents, and $\mathbf{b}$(t) is a vector of independent time-varying current sources and inductor voltages.

Although dynamic solution of equation (1) is useful for transient study and package/decoupling evaluation[9], it is expensive and unnecessary for optimizing pad allocation. Assuming a sound decoupling capacitance design which will cater adequately to transient peak current demands, the pad currents change at a rather low frequency compared to the clock frequency, following more closely the change in chip's power consumption level due to change in the stream of instructions executed[9]. This is so also due to the high parasitic inductance in the board and package. Thus, it is imperative that the dynamic voltage levels in a power grid remain close to the DC voltage levels at a given instant, in order for a chip to perform reliably. In view of this, the pad design problem can be more efficiently tackled by considering different "long-term" (i.e. DC) current demand scenarios for the design, rather than considering transient current behaviors. Our proposed method considers a set of specified DC current distributions in the design corresponding to different chip operating modes, and finds an optimal set of pads that will work well in all these load scenarios. As a result, the optimization problem is concerned with a simpler static model:

$$\mathbf{G} \cdot \mathbf{v} = \mathbf{i}, \tag{2}$$

where $\mathbf{G}$ is a conductance matrix, $\mathbf{v}$ is a vector of node voltages, and $\mathbf{i}$ is a vector of load currents and Norton currents of voltage sources.

Equation (2) can provide the basic constraints for our optimization, should we choose *all* node voltages as constrained variables. Since a power grid has millions of nodes, constraining voltage of every node will increase the complexity of the problem. The main idea is, therefore, to constrain voltages of only a small fraction of nodes. These nodes, called the *observation nodes*, should then be selected in such a way that guaranteeing certain minimum (or maximum) voltage level for them should guarantee *nearly* equal or better voltage levels for *all* other nodes. Once such a selection is made, all nodes except the pad candidates and observation nodes can be abstracted away using the macromodeling idea originally proposed in [2]. The transfer characteristics of a macromodel (schematic shown in Figure 1) are given by

$$\mathbf{I} = A \cdot \mathbf{V} + \mathbf{S}, \qquad \mathbf{I}, \mathbf{V}, \mathbf{S} \in R^m, \qquad A \in R^{m \times m} \tag{3}$$

where m is number of ports in the model, A is a conductance matrix, $\mathbf{V}$ is a vector of port voltages, $\mathbf{I}$ is a vector of currents flowing into the model through the ports, and $\mathbf{S}$ is a vector of currents from each port to the reference node. $\mathbf{S}$ essentially has the effect of moving all current sources connected at internal nodes to the ports of the multi-port macromodel. In our formulation, the ports of the model would consist of pad candidates and observation nodes only.

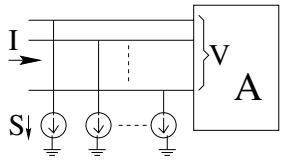

Figure 1: Schematic of a macromodel

The macromodel in equation (3) is derived from modified nodal equations of the network, given by:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} -\mathbf{J_1} \\ -\mathbf{J_2} + \mathbf{I} \end{bmatrix} \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are voltages of internal nodes and ports respectively, $\mathbf{J_1}$ and $\mathbf{J_2}$ are current sources connected at internal nodes and ports respectively, $\mathbf{I}$ is vector of currents through the interface, and G_{ij} are partitions of the admittance matrix G.

From (4), we can derive:

$$A = G_{22} - G_{12}^T G_{11}^{-1} G_{12}, \qquad \mathbf{S} = \mathbf{J_2} - G_{12}^T G_{11}^{-1} \mathbf{J_1} \tag{5}$$

The above computation can be made more efficient without explicitly inverting G_{11}[2]. The idea of using macromodels allows replacing a very large set of constraints (2) by a much smaller set of constraints (3) and thus makes the problem easier to solve.

3. PAD OPTIMIZATION USING MILP

The overall flow of the proposed technique is outlined below.

Step 1 Determine candidate locations for pads in one of the following ways:

- Specified by user, typically a chip integration engineer.
- By considering all possible pad positions on the peripheral power ring
- By considering all possible C4 bump locations
- By considering terminal points on power stripes reaching the design's boundary

Step 2 Simulate the network with an initial, user-determined pad configuration to select a set of *observation nodes*.

Since the optimization will be done guaranteeing a specified voltage level *only* for the observation (and pad candidate) nodes, it is important that they represent adequately all points in the power grid that are likely to see worse voltages. As current sink nodes suffer the largest voltage drop, we limit the selection to among current sink nodes only. A representative set is constructed by selecting nodes from various regions in the design, each representation coming from the node with the worst voltage in that region. For our experiments, we created these regions by dividing the power grid into a uniform grid.

As the worst voltage locations can shift with change of pad configurations, it is possible that the worst voltage realized with the optimal pad configuration falls short of the target voltage, which is set on the worst voltage nodes with the initial pad configuration. In practice, we found that this difficulty can be overcome by two actions: (i) by selecting a fairly larger observation set and (ii) by setting slightly tighter voltage constraints for the observation nodes so that eventually all nodes meet the target voltage. It may be noted that the run time complexity is not impacted much by the size of the observation set which contributes only continuous variables in the MILP problem.

Step 3 Generate a macromodel of the power network with the candidate pad locations (PC) and the observation nodes (OBS) as ports of the model, formulate the MILP system and find its optimal solution. This step is detailed in following subsections.

3.1 MILP Formulation

Without loss of generality, the problem formulation will be discussed for a Vdd supply grid. Formulation for ground grid is similar. Consider a macromodel constructed with ports consisting of pad candidates, PC and observation nodes, OBS. The optimization problem can be stated as:

$$\begin{array}{ll} \text{minimize} & \text{number of pads, N} \\ \text{subject to} & (i)\ I_i \le I_{thre}, \quad \forall\ i \in PC \\ & (ii)\ V_j \ge V_{thre}, \quad \forall\ j \in PC \cup OBS \\ & (iii)\ I_i, V_j \text{ satisfy equation (3)}, \end{array}$$

where I_{thre} is maximum current allowed through pads and V_{thre} is the specified minimum voltage for any node in the power grid.

3.2 Constraints on Pad Candidates

Let us introduce $0-1$ integer variables z, with $z_i = 1$ denoting that a pad is placed at pad candidate i. This will help to set different voltage and port current constraints for PC nodes depending on whether a pad is connected at the candidate location or not. We can now write the constraints for PC ports as:

$$V_i - V_{dd} \cdot z_i \ge 0 \tag{6}$$
$$V_i \le V_{dd} \tag{7}$$
$$V_i \ge V_{thre} \tag{8}$$
$$I_{thre} \cdot z_i - I_i \ge 0 \tag{9}$$
$$I_i \ge 0 \tag{10}$$

The above assumes that pads are at ideal supply voltage of V_{dd}. In reality, supply to a pad is modeled as a voltage source with a series resistance. This is easily done by connecting an additional resistance to each pad candidate, irrespective of whether it will get a pad or not. For those candidates where no pad may be added, the port voltages are same with and without the additional resistance due to zero current through these ports.

When i is a pad, constraints (6) and (7) together impose $V_i = V_{dd}$. Likewise, constraints (9) and (10) together enforce that the current is zero when i is not a pad, and is non-negative when i is a pad.

3.3 Constraints on Observation Points

By further partitioning the macromodel in equation (3) based on PC and OBS ports, we can rewrite it as:

$$\begin{bmatrix} \mathbf{I_{PC}} \\ \mathbf{I_{OBS}} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \mathbf{V_{PC}} \\ \mathbf{V_{OBS}} \end{bmatrix} + \begin{bmatrix} \mathbf{S_{PC}} \\ \mathbf{S_{OBS}} \end{bmatrix} \tag{11}$$

where $\mathbf{I_{PC}}$ and $\mathbf{I_{OBS}}$ are currents through the PC and OBS ports respectively, and $\mathbf{S_{PC}}$ and $\mathbf{S_{OBS}}$ are constant current sources from these ports to the reference node. It should be noted that all elements in $\mathbf{I_{OBS}}$ are zero since there is no current flow into the model through the observation nodes. Based on this, we can derive the following 2 equations:

$$\mathbf{I_{PC}} = A' \cdot \mathbf{V_{PC}} + \mathbf{S'} \tag{12}$$
$$\mathbf{V_{OBS}} = T \cdot \mathbf{V_{PC}} + \mathbf{B} \tag{13}$$

where $\mathbf{I_{PC}}, \mathbf{V_{PC}}, \mathbf{S'} \in R^m$, $\mathbf{V_{OBS}}, \mathbf{B} \in R^n$, $A' \in R^{m \times m}$, $T \in R^{n \times m}$, m is size of PC, n is size of OBS, and

$$A' = A_{11} - A_{21}^T \cdot A_{22}^{-1} \cdot A_{21} \tag{14}$$
$$\mathbf{S'} = \mathbf{S_{PC}} - A_{21}^T \cdot A_{22}^{-1} \cdot \mathbf{S_{OBS}} \tag{15}$$
$$T = -A_{22}^{-1} \cdot A_{21} \tag{16}$$
$$\mathbf{B} = -A_{22}^{-1} \cdot \mathbf{S_{OBS}} \tag{17}$$

The equality constraints (13) can be combined with constraints specifying the threshold voltage V_{thre} at the observation points to generate:

$$T \cdot \mathbf{V_{PC}} \ge \mathbf{C} \tag{18}$$

$$where \quad \mathbf{C} = \begin{bmatrix} V_{thre} - B_1 \\ V_{thre} - B_2 \\ \\ V_{thre} - B_n \end{bmatrix} \tag{19}$$

3.4 Complete Formulation

Eliminating pad current variables from (12), (9) and (10), and combining them with (6)-(8) and (18), we get the complete problem formulation as below:

$$\begin{array}{lll} \text{minimize} & \sum_{i \in PC} z_i, \quad z_i \in \{0,1\} & \\ \text{subject to} & T \cdot \mathbf{V_{PC}} \ge \mathbf{C} \quad and & (20) \\ \forall\ \text{i} \in \text{OBS}: & V_i - V_{dd} \cdot z_i \ge 0 & (21) \\ & V_i \le V_{dd} & (22) \\ & V_i \ge V_{thre} & (23) \\ & I_{thre} \cdot z_i - \sum_{j \in PC} (A'_{ij} \cdot V_j) - S'_i \ge 0 & (24) \\ & \sum_{j \in PC} (A'_{ij} \cdot V_j) + S'_i \ge 0 & (25) \end{array}$$

A problem with m PC and n OBS nodes uses m 0-1 integer variables, m continuous variables, and $n + 5m$ linear constraints. It is worthwhile to note that increasing the number of observation points increases only the number of constraints, but not number of variables. Since the performance of MILP hinges mainly on the number of integer variables, increasing the number of observation points does not affect the efficiency of this procedure much.

3.5 Extension for Multiple Loads

The above formulation is for a single load current distribution in the design. However, it can be extended to handle multiple current distributions by replicating the voltage variables and all the constraints, once for each load current scenario. The optimization problem with p load cases will thus require m 0-1 integer variables, $m.p$ continuous variables, and $p(n+5m)$ constraints. Again, since the number of integer variables which determines the performance of MILP, is independent of the number of load scenarios, the run time complexity of the optimization is not impacted severely. Due to increased constraints, the optimizer will have to work harder to find the optimal solution. The only constant vectors that change with the change in load currents are $\mathbf{S}'$ and $\mathbf{B}$. Multiple computations of $\mathbf{S}'$ and $\mathbf{C}$ are done very efficiently using the pre-factored Cholesky factors of the admittance matrix and forward, backward substitutions, as described in [2]. Thus the overhead of model computation for considering multiple load scenarios is only small.

4. HEURISTIC ALGORITHMS

Branch-and-bound is a widely used technique in 0-1 MILP. We found it necessary to improve the performance of the optimizer through additional heuristics, which cut down the number of integer variables.

4.1 Pruning of Pad Candidates

During macromodeling, the internal currents are redistributed among the ports as current vector S depending on the conductance between internal nodes and ports. A port, i, with high conductance to internal nodes will get more current (larger S_i). If no pad is attached to certain port, i, then the current S_i has to flow into the macromodel from ports that have pads, and out to the port, i. Intuitively, this suggests that it is better to connect pads at ports with large S_i, as otherwise large currents will have to flow in and out of the model. That is, we would want to connect pads to ports that have large conductance to nodes with internal current sources.

We iteratively prune out a small set of pad candidates with the smallest S_i and recalculate the new $\mathbf{S}$ vector for the remaining ports. $\mathbf{S}$ should be recomputed as often as practical, as the port current distribution changes when some ports are eliminated. There may be cases (somewhat pathological) where some internal nodes with a small total current have high conductivity to only one pad candidate, call it k, whereas k itself is not connected tightly to other pad candidates. In this case, it may be desirable that a pad is assigned at k, as otherwise those internal nodes may see severe voltage drop. There is a risk of candidate k being eliminated by the pruning heuristic due to the expected small current, S_k. To handle such situations better, we perform pruning in a modified way, as per the following procedure.

1. Partition PC set into tightly connected sets (node graphs with high conductance edges) based on the values of A_{ij}.
2. Based on the total number of pad candidates desired after pruning, budget the number of pad candidates for each partition based on the ratio of total S_i currents in each partition. Ensure also that the budget is at least as big as the number of pads required to meet the total current in that partition without violating the maximum pad current limit.
3. Prune candidates for each partition independently to meet the PC budget for that partition.

4.2 Divide and Conquer

When the initial PC set is large, aggressive pruning is required to reduce the problem size, but aggressive use of the procedure in Section 4.1 may sometimes result in poor quality solutions, or render the problem infeasible. So, we devised another approach that will reduce the problem size by subdividing the problem, but without pruning. The key idea is to divide the PC and OBS sets into partitions of manageable PC size, and then do pad assignments successively for one partition after another, considering *only* the PC set in each partition, and with the *limited* goal of guaranteeing the target voltage only for the small OBS set in that partition.

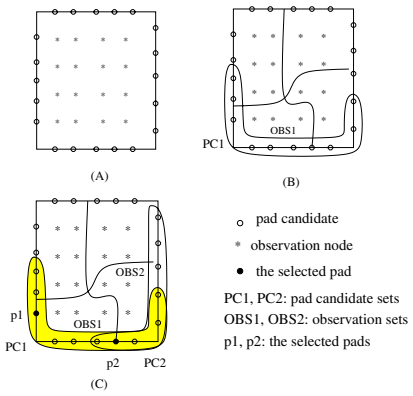

Figure 2: Illustration of Divide-and-Conquer

The procedure is illustrated through an example in Figure 2. The network consists of 20 PC ports along the periphery (marked 'o') and 16 OBS ports (marked '*'). Through a clustering procedure (explained later), a partition consisting of a small subset of PC, marked $PC1$, and a subset of OBS, marked $OBS1$, is created. The pad optimization problem is now solved with $PC1$ and $OBS1$ as ports and targeting to meet the specified voltage for $OBS1$ nodes. Suppose this step assigned pads to candidates $p1$ and $p2$. Now, the next partition consisting of $PC2$ and $OBS2$ is constructed. This partition may share pad candidates with partitions solved already. If any of the shared nodes already got a pad (eg. $p2$ in the illustration), that node is treated as pad while optimizing the newer partition. This procedure is continued until all the OBS nodes are covered.

The algorithm, given as pseudocode in Figure 3, constructs a partition by adding alternatingly OBS and PC nodes to the partition. Starting with a randomly chosen OBS node as the first entry into the partition, PC nodes which have significant influence on the voltage of the chosen OBS node are pulled in next. Based on equation (13), a pad candidate j is considered to have significant influence on observation node i, if matrix entry T_{ij} is above a preset threshold. While admitting PC nodes to the partition, their influence (called

weight in the pseudocode) on other OBS nodes that are not currently in the partition are tracked using a cumulative T_{ij} measure. When no more PC node can be added to the partition, either because the maximum pad candidate limit for the partition has been reached, or because no more pad candidate has significant influence on the recently added OBS node, a new OBS node with the largest weight is admitted to the partition. The above steps are repeated until maximum pad candidate limit is reached or all observation nodes have been covered.

Input: PC, OBS (initial pad & observation node sets)
N_{max}, maximum size of PC per MILP solve
T_{thre}, value of $T_{i,j}$ below which an OBS port is considered unrelated to a PC port
Output: $S \subset OBS$, $P \subset PC$ s.t. $|P| \leq N_{max}$
$LostValue$ array for adjusting V_{thre}

1. Put a randomly chosen OBS port, i, into S
2. while $|P| < N_{max}$
3. for each $j \in PC - P$
4. if $T_{i,j} \geq T_{thre}$
5. Put j into P
6. for each $k \in OBS - S$: $weight(k) += T_{k,j}$
8. else $LostValue(i) += T_{i,j}$
9. if $|P| < N_{max}$
12. Put $i \in OBS - S$, whose *weight* is the largest, into S

Figure 3: Algorithm for clustering OBS and PC

Note that the algorithm also tracks a $LostValue$ for each selected observation node. This value represents the error caused by neglecting the influence of pad candidates that have not been selected for the current partition. Since the voltage of any pad candidate outside the current partition is expected to be in the range $[V_{thre}, V_{dd}]$, the error in the voltage of the observation node i due to excluding those pad candidates from the partition will be in the range of $[V_{thre} \cdot LostValue(i), V_{dd} \cdot LostValue(i)]$. So, we can conservatively make up for the error in $\mathbf{C}$ by replacing equation (19) by

$$\mathbf{C} = \begin{bmatrix} V_{thre} - B_1 - LostValue_1 * V_{thre} \\ V_{thre} - B_2 - LostValue_2 * V_{thre} \\ - - - - - - - - \\ V_{thre} - B_n - LostValue_3 * V_{thre} \end{bmatrix}.$$

The clustering procedure is illustrated in Figure 4. An example T matrix is shown in Figure 4(A), and two partitions obtained from T are shown in Figure 4(B). In this example, pad candidates $p4$ and $p5$ have little influence on the voltage of observation nodes $s1$ and $s2$, whereas $p3$ has significant impact on all observation nodes. A bi-partition graph consisting of all PC and OBS candidates can be obtained by deleting small terms in T and drawing edges between PC and OBS nodes corresponding to significant T_{ij} entries. Assuming N_{max} is 3, the complete bi-partition graph will be partitioned into 2 bi-partition graphs. Stepping through the algorithm in Figure 3, one can see the order in which nodes will be added to these partitions. The order will be $\{s1, p1, p2, p3, s2\}$ for the first partition, and $\{s3, p3, p4, p5, s4\}$ for the next partition.

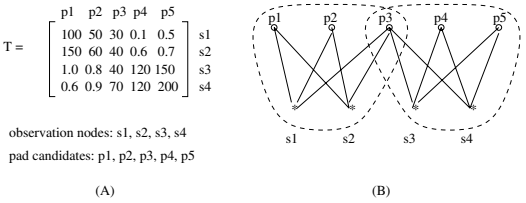

Figure 4: An illustration of the clustering procedure

5. EXPERIMENTAL RESULTS

The proposed techniques were implemented as part of an existing in-house power grid analysis tool[1] using a public domain MILP solver, GLPK[10] and bench-marked using power networks from 5 real designs (see Table 1). In column 1 of the table, chip-1 is top-level power grid of a high-performance processor, chip-2 and chip-3 are micro-controllers, chip-4 is a DSP processor, and memory-1 is a compiled memory module. Columns 2 and 3 show the pad/pin count and the worst voltage obtained with the original unoptimized pad configuration, and columns 4 and 5 show the number of pad candidates and observation nodes selected for optimization. Column 6 shows the worst voltage obtained when *all* pad candidates were serviced with pads, thus providing an upper bound on the worst voltage obtainable with any pad configuration. Columns 7-11 show the results with optimized pad configuration. Column 7 gives the target voltage specified, column 8 the optimal number of pads, and column 9 the actual worst voltage realized with the optimal pad configuration. Column 10 shows the run time (on a 400MHz SUN Sparc workstation) and column 11 the heuristic methods used in solving the MILP. Here, methods 1, 2, and 3 refer respectively to MILP optimization using branch-and-bound, candidate pruning approach described in Section 4.1 and the divide-and-conquer method described in Section 4.2.

Voltage reported in column 9 is the worst voltage among all nodes in a design, not merely among the observation nodes, and was obtained from a simulation with optimized pad configuration. It can be seen that the globally worst voltages are equal or better than the specified voltages in 19 out of 22 cases, verifying the adequacy of our observation points selection. It may be noted that very few (< 1000) observation points are able to effectively capture the worst IR drop in the entire design. Only in very few cases, the actual worst voltage deviated marginally, and it is easy to provide for this by specifying a slightly tighter voltage constraint.

A comparison of columns 8 and 2 (optimal and original pad counts) shows effectiveness of the method in reducing pad count to reach a voltage target, notably in the case of the memory module where the pin count was reduced from 1963 to 10. This result is not surprising if we note that the memory module has a very dense grid in the upper layers and that the original pins had been placed nearly on every horizontal and vertical power stripe in the top two layers.

Figure 5 shows the worst voltage values obtained with various number of pads, using the same data as Chip-1 in Table 1. This illustrates the 'law of diminishing return' as more pads are added to a design. When there are fewer pads, any additional pad reduces the IR drop very effectively. But this effectiveness decreases as more and more pads are added to the design.

The CPU run time (column 10) includes the time taken for preparing and processing the macro-models, and MILP optimization. It took less than an hour for any of the benchmarks. The divide-and-conquer heuristic (method 3) is very useful in reducing the run time significantly (eg. chip-3) or finding higher quality solutions (eg. chip-1, memory-1) or both (eg. chip-1).

Table 2 shows results of optimization considering multiple power demand scenarios. It compares results based on 5 and 10 load patterns with that of single load pattern. Columns 3,

Table 1: Benchmark Results of Pad Optimization

Network (#Nodes)	Original Design		With maximum pads			After Pad Optimization				
	#Pads/ #pins	Worst Vol.(V)	#PC	#OBS	Worst Vol.(v)	Spec Vol.(v)	#Pad	Worst Vol.(v)	CPU(s)	Heuristics
Chip-1 (93K nodes)	61	2.47	79	48	2.47	2.30	4	2.30	10	1,2
						2.40	7	2.40	13	1,2
						2.43	10	2.43	770	1,2
						2.47	27	2.47	37	1,2,3
Chip-2 (4.9M nodes)	30	2.76	78	132	2.76	2.30	12	2.41	196	1,2
						2.50	12	2.50	196	1,2
						2.70	13	2.67	200	1,2
						2.76	14	2.76	200	1,2
Chip-3 (294K nodes)	4	1.38	42	295	1.44	1.35	3	1.36	29	1,2
						1.39	4	1.38	51	1,2
						1.41	5	1.41	67	1,2
						1.42	7	1.42	198	1,2
						1.44	21	1.44	27445	1,2
						1.44	22	1.44	752	1,2,3
Chip-4 (3.9M nodes)	4	2.81	42	595	2.93	2.80	3	2.80	121	1
						2.85	4	2.85	121	1
						2.88	5	2.88	132	1
						2.90	6	2.90	134	1
						2.93	13	2.93	159	1,2
Memory-1 (1.6M nodes)	1963	1.28	2144	90	1.30	1.25	1	1.25	90	1,2
						1.28	4	1.28	533	1,2
						1.30	10	1.29	3382	1,2,3

Table 2: Optimization over Multiple Load Patterns

Network	Spec Vol (v)	Single load pattern			5 load patterns			10 load patterns		
		#Pads	CPU(s)	#iter	#Pads	CPU(s)	#iter	#Pads	CPU(s)	#iter
Chip-1	2.40	7	798	172855	8	569	31415	9	1086	29694
Chip-3	1.35	3	330	31116	3	1387	55871	3	20115	410433
Chip-4	2.85	4	121	663	7	1119	8419	Infeasible		

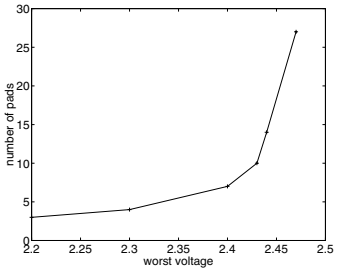

Figure 5: Effect of pad count on worst voltage

6, and 9 show the optimal pad counts, columns 4, 7, and 10 show the run time, and columns 5, 8, and 11 show the number of branch-and-bound iterations. The multiple load patterns were mimicked by altering the power of all cells and circuit blocks randomly within ±30% of the original power dissipation. The results show that only few additional pads are required to satisfy multiple load cases. The optimizer had to work harder under tighter constraints imposed by multiple load cases, explaining the longer run times in those cases. In case of chip-4, the specified 2.83V was infeasible even if all pad candidates are serviced. Note that Table 2 results were obtained without using the additional heuristics of Section 4, though we could have applied them if it were necessary. This explains the longer run times for some cases in Table 1 than in Table 2 for the single load cases.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented several techniques for efficiently selecting locations for power supply pads, pins, or voltage regulators. The problem was formulated as a mixed integer linear program optimization using macro-modeling techniques to reduce the problem complexity. Moreover, two heuristic techniques were presented to further reduce the computational complexity of this problem. Benchmark results on several real designs demonstrated the effectiveness of the proposed techniques. Extensions for obtaining optimal pad configurations satisfying voltage requirements across multiple chip loading conditions were also presented and validated experimentally.

[The authors would like to thank Prof. Jiang Hu, Texas A&M Univ. for suggesting the divide-and-conquer heuristic.]

7. REFERENCES

[1] A. Dharchoudhury, et. al. "Design and analysis of power distribution networks in PowerPC microprocessors," in *DAC*, pp. 738–743, 1998.

[2] M. Zhao, et. al. "Hierarchical analysis of power distribution networks," *IEEE Trans. on CAD*, vol. 21, pp. 159–168, Feb. 2002.

[3] T. Mitsuhashi and E. S. Kuh, "Power and ground network topology optimization for cell based vlsis," in *DAC*, 1992.

[4] X. D. Tan, et. al. "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings," in *DAC*, pp. 78–83, 1999.

[5] X. Wu, et. al. "Area minimization of power distribution network using efficient nonlinear programming techniques," in *ICCAD*, pp. 153–157, 2001.

[6] T. Y. Wang and C. P. Chen, "Optimization of the power/ground network wire-sizing and spacing based on sequential network simplex algorithm," in *ISQED*, pp. 157–162, 2002.

[7] K. Wang and M. M. Sadowska, "Power/ground mesh area optimization using optimization using multigrid-based technique," in *DATE*, 2003.

[8] J. Oh and M. Pedram, "Multi-pad power/ground network design for uniform distribution of ground bounce," in *DAC*, pp. 157–162, 1998.

[9] R. Panda, et.al., "Model and analysis for combined package and on-chip power grid simulation," in *ISLPED*, pp. 179–184, 2000.

[10] "GNU Linear Programming Kit Users' Guide," 2001.