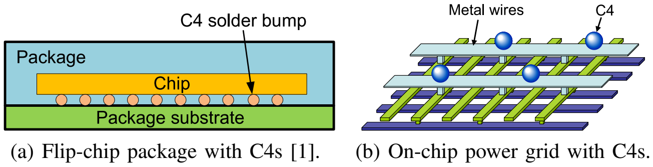
Placement Optimization of Power Supply Pads Based on Locality

Pingqiang Zhou, Vivek Mishra, and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA.
{pingqiang, vivek, sachin}@umn.edu

***Abstract*—This paper presents an efficient algorithm for the placement of power supply pads in flip-chip packaging for high-performance VLSI circuits. The placement problem is formulated as a mixed-integer linear program (MILP), subject to the constraints on mean-time-to-failure (MTTF) for the pads and the voltage drop in the power grid. To improve the performance of the optimizer, the pad placement problem is solved based on the divide-and-conquer principle, and the locality properties of the power grid are exploited by modeling the distant nodes and sources coarsely, following the coarsening stage in multigrid-like approach. An accurate electromigration (EM) model that captures current crowding and Joule heating effects is developed and integrated with our C4 placement approach. The effectiveness of the proposed approach is demonstrated on several designs adapted from publicly released benchmarks.**

I. Introduction

Power delivery is a key performance bottleneck in high-performance integrated circuits. In modern designs, power integrity is ensured by providing power through a large number of package pins. In advanced designs, flip-chip packaging, where connections are provided all over the chip area, has supplanted wire-bonded packing, where pins are only available on the periphery, due to its ability to provide much larger pad counts. Figure 1(a) shows a schematic of a flip-chip package, where Controlled Collapse Chip Connection (C4) solder bumps located throughout the die area are used to connect the die to the package. The view from the chip is shown in Figure 1(b), where we see the C4 locations in the on-chip power grid, relative to the top metal layers of the chip.



(a) Flip-chip package with C4s [1]. (b) On-chip power grid with C4s.

Fig. 1: C4 bumps.

The number of C4s required for a chip can be as high as hundreds or even thousands, depending on the size, power consumption of the design, and the design of the power grid [2]. The problem of power supply pad placement, or C4 placement, determines which C4s are used for power delivery and aims to find the layout, i.e., the number and positions of the C4s over the chip area. Due to increasingly stringent performance requirements on the power delivery network, and the large design space associated with the large number of candidate locations, C4 placement has become difficult and the traditional approach of manual C4 placement is not sustainable. Therefore, a CAD solution is required to find the best C4 layout for a given power domain, capturing the requirements and constraints for power delivery.

The location of C4s affects power integrity in several ways. First, broadly speaking, more C4s are required in areas of high current demand in order to reduce on-chip losses and maintain appropriate voltage levels on the supply and ground networks. Second, it has been projected that C4 solder bump reliability will become increasingly important in future IC designs [3]. C4 bumps are susceptible to electromigration (EM) failure [4]–[6], caused by issues such as:

- Joule heating: When current flows through a conductor (metal wire or C4 bump), the electrons (charge carriers) collide with the metal atoms and produce heat energy, creating a self-heating effect in the wire or bump that causes a local temperature increase.
- Current crowding: When the current flows from a C4 bump to the metal wire on chip, the current density increases because the cross section of the metal wire on the chip side is smaller than the solder bump by at least two orders in magnitude [7]. This causes an effectively larger current density than the case where the current is evenly distributed over the bump.

The C4 pad optimization problem has been investigated by prior work [2], [8]–[10]. In [8], a min-forest heuristic is proposed for pad assignment and power routing on power/ground tree. The work in [9] proposes a greedy approach to iteratively add power pads to the chip one by one, while minimizing the worst voltage drop in the power grid in each iteration. In [10], a simulated-annealing based algorithm is proposed to search the pad placement that minimizes both the worst voltage drop and the voltage variation across the power grid.

Zhao et al. [2] model the power supply pad placement problem as a mixed integer linear program (MILP) (a similar idea has been explored in [11] for the placement of voltage regulators on chip). The approach uses an integer 0-1 decision variable for each pad candidate and formulates the problem using linear constraints based on macromodeling techniques [12]. While the MILP formulation is able to find the optimal solution to the problem, it is NP-hard [13] and can be computationally expensive when the number of integer variables, which equals to the number of pad candidates in the formulation, grows large. This limitation arises because MILP problems are combinatorial optimization problems with an exponential number of feasible points, and so a practical method can only use MILPs of small sizes.

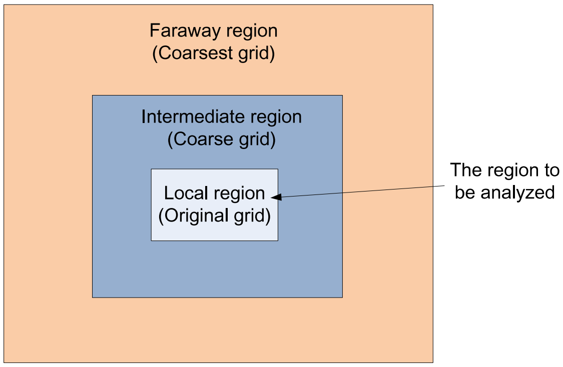


Fig. 2: The concept of locality in power grid and the idea of using coarse grids for remote regions in C4 placement.

This paper presents an efficient and scalable MILP-based algorithm to optimize the layout of power pads on chip, subject to constraints on the voltage drop in the power grid and the mean-time-to-failure (MTTF) for the C4 bumps. To improve the performance of the optimizer, our algorithm exploits the locality properties of the power grid, which imply that most of the current drawn at a node in the power grid originates at nearby power pads [14], [15]. This

978-3-9815370-0-0/DATE13/©2013 EDAA.

aspect of our approach is sketched in Figure 2. The local region, which is close to the area being analyzed, is modeled exactly, but more distant parts of the power grid (labeled as the intermediate and faraway regions) are modeled increasingly more coarsely with their distance from the area being analyzed. Our coarsening strategy resembles the coarsening phase of the multigrid approach [16]–[18]. Specifically, we use a *divide-and-conquer* approach to partition the chip into smaller regions/partitions, and then formulate a small MILP for each local region that includes the original C4 candidates in that region and *lumped* C4 candidates in the coarse grid of the rest chip.

The contributions of this paper are summarized below:

- We develop a divide-and-conquer-based approach to build a *fast, scalable* MILP-based C4 pad optimizer, exploiting the concept of locality to model various regions of the power grid at appropriate levels of coarseness. We use a quad-tree based partition for the whole power grid and solve a series of small MILP problems, one for each partition, preserving reasonable accurate information of the remote regions by including their coarse grids with reduced number of C4 candidates.
- We integrate a temperature-dependent EM model that accurately captures the effects of current crowding and Joule heating on the C4 pads into our MILP-based C4 placement algorithm. In contrast, prior work [2] models EM using simple constraints that assume that the maximum current through a C4 is temperature-independent. Our experimental results show that the variance in maximum current through a C4 across the chip can be significant (more than 50%). Moreover, we show that using temperature-independent current limit for the C4s will lead to excessive resource usage.

The rest of the paper is organized as follows. In Section II we present the overview of the proposed C4 placement algorithm. In Section III, we describe the approach to build coarse grids. We discuss the temperature-dependent EM model in Section IV, and then present the MILP formulation to solve the C4 placement problem in Section V. The efficiency of our proposed algorithm is verified by the experimental results given in Section VI. Finally, conclusions are made in Section VII.

II. The Overall Optimization Flow

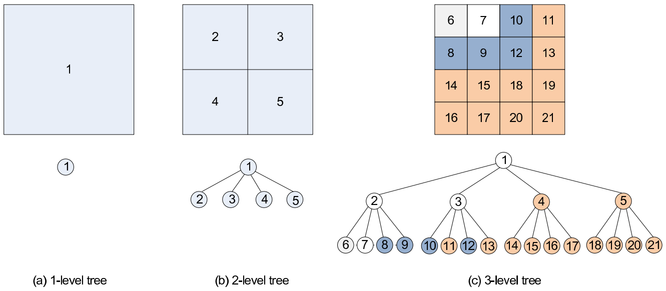
A. The Quad-Tree Representation of the Layout

We apply a *divide-and-conquer* approach to our C4 placement procedure, based on dividing the chip area into smaller regions or partitions. We solve the C4 placement problem in each partition using the notion of locality in the power grid.

We superimpose a quad-tree structure over the chip area to help create partitions for the power grid at various levels of granularity. The structure of this quad-tree, for three different depths, is illustrated in Figure 3. Each node in the tree corresponds to a partition. At the top level (Figure 3(a)), the entire chip corresponds to the solitary partition, at the next level, it is divided into four parts (Figure 3(b)), then sixteen parts (Figure 3(c)), and so on. Thus, each node in the tree represents a partition of the power grid at a certain level of granularity.

A tree with larger depth has a larger number of partitions to work with, but each such partition has a smaller size of problem to solve. The size of a leaf node should be such that it can be treated as a mostly-independent object, i.e., that a majority of locality effects are contained within the partition. Moreover, the formulated MILP in each partition should be small enough that it can be practically solved in a reasonable amount of time. In our implementation, given a power grid, we build the quad-tree with a reasonable depth in order to consider both of these factors, each facilitated by one controlling parameter:

- a parameter to specify the allowable minimal number of power grid nodes in each leaf node. This parameter would set the upperbound for the depth of the quad-tree, i.e., the minimal size of partition/leaf node.



(a) 1-level tree (b) 2-level tree (c) 3-level tree

Fig. 3: The quad-trees representing different partitioning levels of the same power grid. The leaf nodes are the pending partitions. In the 1-level tree, the single partition 1 represents the whole chip. In the 2-level tree, the chip is partitioned into 4 small partitions. And in the 3-level tree, the chip is composed of 16 smaller partitions.

- a parameter to specify the allowable maximum number of candidate C4s in each leaf node, so as to control the size of MILP. This parameter would set the lowerbound for the depth of the quad-tree.

In reality, these two parameters can be set based on the experience of the designers/users, and a proper level of chip partitioning can be found using these two parameters.

B. Outline of the Algorithm

We begin with a quad-tree partitioning of the chip at the appropriate level, balancing out the tradeoffs described above. Each partition has a designated number of C4 candidates and observation (OBS) nodes, which can be provided by the user. In our implementation, we use the method described in [2] to choose these C4 candidates and observation nodes. The partitions are then processed one by one, as illustrated in Figure 3(c). The figure shows a quad-tree with 16 partitions at the leaf level. We identify three kinds of partitions in the problem formulation:

- An *active* partition that is currently being solved.
- A *finished* partition that has already been solved.
- A *pending* partition that is unsolved so far.

Let us assume that at an intermediate stage of the computation, partition 6 is finished, and all other partitions are pending.

In each iteration, we pick a partition from the list of pending partitions, for example, partition 7 in Figure 3(c), make it the active partition. We then formulate, as described in Section V, the C4 placement problem associated with selecting the C4 assignments in each partition: the currently active partition (here, partition 7), the finished partition(s) (here, partition 6), and all pending partitions (here, partitions 8–21). Note that these partitions are, by definition, non-overlapping, and that their union represents the complete region of the whole power grid.

The C4 locations for the finished partitions are determined at a previous step, and these are no longer optimization variables in the MILP formulation (discussed in Section V) for the currently active partition. However, the *original* OBS nodes for the finished partitions must be represented in the optimization, in order to ensure that the addition of new C4s in the active partition does not degrade the voltages at these observation nodes.

For all pending partitions, the locations and numbers of the C4 candidates are as yet undetermined, and must be modeled as optimization variables. To facilitate this in a computationally efficient way, the partitions other than the active partition are grouped into two classes based on the concept of locality illustrated in Fig. 2:

- *intermediate partitions* that lie in the intermediate region for the active partition, and
- *faraway partitions* that lie in the faraway region relative to the active partition.

For the example in Figure 3(c), partitions 6, 8, 9, 10, and 12 are intermediate partitions relative to partition 7, and partitions 11 and

13-21 are faraway partitions.

Based on this classification of the partitions, the C4 nodes are modeled at various levels of detail. In particular, we use a *lumped* representation for the C4 candidates and OBS nodes in faraway regions by following the coarsening phase of the multigrid approach as presented in Section III. Such a representation represents an approximation that is good enough to capture the effects of faraway C4s (which supply a small, but not zero, fraction of the current to the active partition). In other words, we ensure that the effects of faraway regions are accounted for at just the level of accuracy required, but that computational efficiency is enhanced.

For the example in Figure 3(c), partitions 14–17 are represented using a coarse partition 4, and partitions 18–21 by coarse partition 5, so that the analysis is performed using ten partitions instead of sixteen: partitions 6 through 13 at the leaf-node level, and partitions 4 and 5 at one level above. The use of partitions 4 and 5 in this case results in a reduction of the problem size as compared to a flat representation, using the coarsening method discussed in Section III.

The C4 placement problem for the active partition, using such coarsening, is then formulated as an MILP using a macromodeling technique discussed in Section V-B, with the objective of minimizing the total number of used C4s in the active, pending, and finished partitions, while meeting the IR drop and EM constraints in these partitions. After solving this MILP problem, we fix the C4s used in the active partition. For the C4s used in the pending partitions, we save the solutions for these partitions and then use them as an initial guess (which is requested by our MILP solver) for C4 placement problems in subsequent iterations. We repeat this procedure considering each leaf node of the quad-tree as an active partition, until all the pending partitions are solved.

III. Grid Coarsening

Our approach is based on the idea of using coarsened grids for faraway partitions in order to limit the size of the optimization problem, while maintaining reasonable accuracy. Specifically, this coarsening reduces the number of variables associated with C4 candidates and the constraints associated with OBS nodes, and also reduces the number of general nodes in the power grid.

The original power grid is irregular in the 3-D space (x, y and z directions), because different layers have different wire pitches and wire widths, and vias are used to connect the wires from different layers in the z-direction. In our work, we follow the procedure in [18] to obtain a 2-D (x and y directions) regular grid from a give 3-D power grid. When a fine grid is coarsened to a coarse grid, the horizontal/vertical conductances from the x-/y-directional connections are averaged among every four conductances, respectively, while the current loads are summed up [18].

For the pad conductance of the C4 candidates, we first assume all the C4 conductance is identical, which is a reasonable assumption in the real design (our method can be extended to handle the situation where this is not the case). When a C4 candidate is used, the power grid is connected to the ideal voltage source with a C4 conductance of G_z (which can be obtained from the technology file in real designs), otherwise, the C4 conductance is 0 if the C4 candidate is not used. When we build the coarse grid, the conductance from the C4 candidates in the fine grid is a varying value, depending on how many C4s are used in the coarsened region. It can be as small as 0 if none of these C4 candidates is used, or as large as the sum of conductance from all the C4 candidates if all of them are used, as shown in Figure 4.

On average, four nodes in the fine grid are lumped into one single node in the coarse grid, so the total grid size is reduced by a factor of four. A node is an OBS node in the coarse grid so long as one of the corresponding four nodes in the fine grid is an OBS node.

IV. EM model for C4 Bump

In this section, we present a EM model for the C4 bumps that can be integrated into our C4 placement framework, and also captures the current crowding and Joule heating effects.

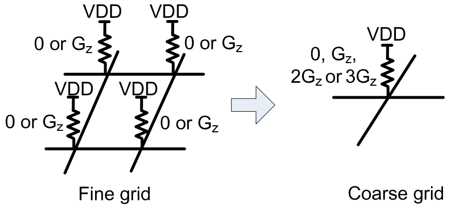


Fig. 4: When building a coarse grid from a fine grid, the valid C4 conductance is determined by the actual number of C4 candidates used.

The mean time to failure (MTTF) of a C4 bump due to electromigration (EM) can be calculated using a modified Black's Equation [4]:

$$\text{MTTF} = A\frac{1}{(cj)^n}\exp\left[\frac{Q}{k(T+\Delta T)}\right] \tag{1}$$

where A is a constant, c is the factor capturing the current crowding effect, j is the average current density, n is a model parameter, Q is the activation energy, k is Boltzmann's constant, T is the temperature at the C4 spot caused by the power consumption of the chip, and ΔT is the local temperature increase due to Joule heating, which can be calculated as [19]:

$$\Delta T = I^2 R R_\theta \tag{2}$$

Here, $I = j \cdot S$ is the current through the C4 bump (where S is the area of C4 bump), R is the C4 resistance and R_θ is the thermal resistance of the C4 bump to the substrate.

Using Equation (2), Equation (1) can be transformed to

$$\text{MTTF} = \frac{AS^n}{c^n I^n}\exp\left[\frac{Q}{k(T+I^2RR_\theta)}\right] \tag{3}$$

This EM model shows that the MTTF of a C4 bump is dependent on I, the current through the C4 and T, the temperature value at the C4 spot; all other terms are constants for a given technology.

Taking the logarithm of both sides of Equation (3), we get

$$\ln(\text{MTTF}) = \ln A + n\ln(S/c) - n\ln I + \frac{Q}{k(T+I^2RR_\theta)} \tag{4}$$

Taking the derivative of $\ln(\text{MTTF})$ w.r.t I, we have

$$\frac{dln(\text{MTTF})}{dI} = -\frac{n}{I} - \frac{2QRR_\theta I}{k(T+RR_\theta I^2)^2} < 0 \tag{5}$$

From the inequality above, we can conclude that $MTTF$ is a strictly decreasing function of I. To ensure that $MTTF \geq \text{MTTF}_{min}$, the specified design objective of $MTTF$, we only need to bound the C4 current, I, as

$$I \leq I_{max}(T) \tag{6}$$

where $I_{max}(T)$ is the current at which MTTF is equal to MTTF_{min}. It is determined by the temperature T, and can be obtained by solving Equation (3). However, since the equation does not admit a closed-form solution, we build a look-up table to characterize $I_{max}(T)$ for various values of MTTF_{min} and temperature T. Based on a thermal analysis of the chip, the value of T for each solder bump can be determined, and the lookup table yields the constraint on the right hand side of Equation (6).

V. MILP Formulation

In this section, we show that the C4 placement problem as described in Section II can be formulated as series of small MILPs. We first, in Section V-A, present a method to reduce the number of 0-1 variables that represent the number of C4 candidates in the problem, then macromodel the power grid in Section V-B, and finally present the complete MILP formulation in Section V-C.

Our optimization formulation differs from previous work [2] in that

- we explicitly consider the C4 conductance in the optimization formulation, and
- we present an MILP formulation for the circuit with lumped C4s in the coarse grids whose valid conductance varies with the actual number of C4 bumps used.
- we incorporate an EM model that captures the effects of current crowding and temperature.

A. Choice of 0–1 Variables

In the active grid, the MILP must determine the number and locations of the inserted C4s, while in the coarsened grids, the MILP must represent or determine how many C4s are to be inserted. We now consider the representation of the 0–1 variables that represent the number of C4s in the coarsened grids.

A C4 port i with a maximum of l_i available C4 bumps may have a conductance of $q_i \cdot G_z$, where q_i is an integer that takes a value in the interval $[0, l_i]$, and represents how many bumps are actually used. To transform this to a 0–1 MILP, one possible way is to introduce l_i new 0–1 variables, each corresponding to a C4 pad; the sum of these variables is then q_i, the number of C4s actually used.

Instead, to reduce the number of variables, we model this using a set of P_i C4 conductors connected in parallel at port i, where $2^j \cdot G_z$ is the conductance of the j^{th} conductor, and

$$P_i = \lceil log_2(l_i + 1) \rceil$$

is the minimum number of bits to represent the integer number l_i in binary mode. We can now represent the number of used C4s in port i using a set of 0–1 integer variables y_{ij}, and the number of C4s actually used is given by

$$q_i = \sum_{j=0}^{P_i-1} y_{ij} \cdot 2^j$$

.

Note that this transform is applicable to a partition at any level of coarseness, and therefore provides a unified formulation for the quad-tree partitions of different levels of coarseness. For a partition with original/flat power grid, we have $l_i = P_i = 1$ since the C4 nodes are not agglomerated, and the number of variables is unchanged. However, for the lumped C4 ports in the coarse grid, where $l_i \geq 2$ (typically much larger, up to a few hundred), this method brings down the number of 0–1 variables y_{ij}s from O(l_i) to O($\log_2 l_i$).

B. Macromodeling of the Power Grid

The power grid obtained after assembly of the partitions, as presented in Section II, may have millions of nodes, but our analysis is only required to monitor OBS, the set of n selected observation nodes for the power grid, and Src, the m predefined candidate connection nodes for the C4 bumps. Therefore, we build a macromodel whose ports are these $n + m$ nodes, and abstract away all of the other nodes in the network using the macromodeling approach [12]. Figure 5 shows the macromodel of the power grid. Note that the C4 ports and OBS ports are a mix of the C4 candidates and OBS nodes from the original power grids in the finished partitions or active partition, with the lumped C4 candidates and OBS nodes from the coarse grids in the pending partitions.

The equations for DC analysis of a power grid are [20]:

$$MX = E \tag{7}$$

where M is the conductance matrix, X is the voltage vector of nodes in the power grid, including both the port nodes and the internal nodes, and E is the vector of current loads. From Equation (7), we can obtain the macromodel of the power grid that includes only the C4 and OBS ports as [2]:

$$I = DV + S \tag{8}$$

where I is a vector of currents flowing into the system through the ports, D is the admittance matrix, V is the vector of voltages only for the ports, and S is a vector of currents from each port to the reference node (see Figure 5).

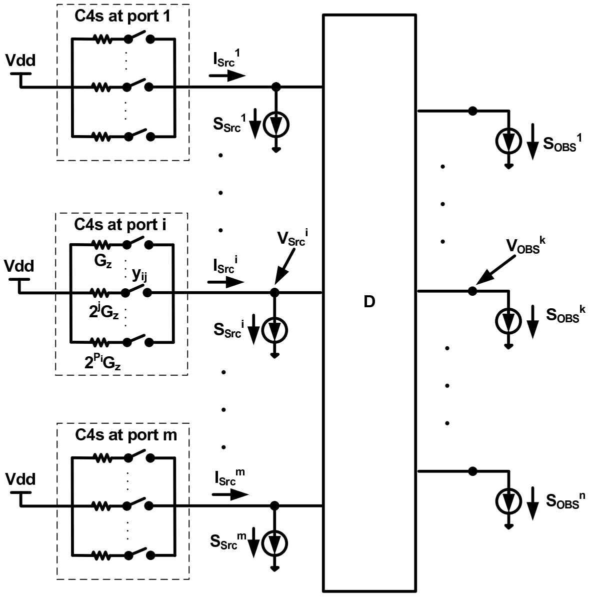


Fig. 5: Macromodel of the power delivery network after assembling all the valid partitions.

If we further partition the ports into two sets, Src and OBS, Equation (8) can be rewritten as

$$\begin{bmatrix} I_{Src} \\ I_{OBS} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} V_{Src} \\ V_{OBS} \end{bmatrix} + \begin{bmatrix} S_{Src} \\ S_{OBS} \end{bmatrix} \tag{9}$$

where I_{Src}, I_{OBS}, V_{src} and V_{OBS} are respectively the current and voltage vectors of the Src and OBS ports. Since the current flowing into the system only through the C4 ports, i.e., $I_{OBS} = 0$, we have:

$$V_{OBS} = WV_{Src} + U \tag{10}$$

where $W = -D_{22}^{-1}D_{21}$, and $U = -D_{22}^{-1}S_{OBS}$. Further,

$$I_{Src} = D_{11}V_{Src} + D_{12}V_{OBS} + S_{Src} = D'V_{Src} + S'_{src} \tag{11}$$

where $D' = D_{11} + D_{12}W$ and $S'_{src} = S_{Src} + D_{12}U$.

Equations (10) and (11) tell us that the current vector of the Src ports I_{Src} and voltage vector of the OBS ports V_{OBS} are linear functions of the voltage vector of the Src ports V_{Src}. This allows us to propose the MILP formulation in Section V-C.

C. Complete MILP Formulation

Using the macromodel shown in Figure 5, the optimization problem is to find the optimal y_{ij} assignments, while meeting EM constraints for the C4 bumps and the IR drop constraints for the OBS nodes.

Let $C4_{fixed}$ be the set of C4 ports in the finished partitions. The C4 placement can be formulated as shown below (a detailed explanation of every equation follows the statement of the optimization problem):

$$\text{minimize} \quad \sum_{i=1}^{m} \sum_{j=0}^{P_i-1} y_{ij} \cdot 2^j \tag{12}$$

subject to

$\forall k \in$ OBS:

$$V_{OBS}^k = \sum_{i=1}^{m} (W_{ki} \cdot V_{Src}^i) + U^k \geq V_{th}^k \tag{13}$$

$\forall i \in$ Src:

$$\sum_{j=0}^{P_i-1} y_{ij} \cdot 2^j \leq l_i \tag{14}$$

$$I^i_{Src} = \sum_{j=1}^{P_i} I^{ij}_{Src} = \sum_{k=1}^{m} (D'_{ik} \cdot V^k_{Src}) + S'^{i}_{Src} \tag{15}$$

$$0 \le I^{ij}_{Src} \le (I^i_{max}(T_i) \cdot 2^j) \cdot y_{ij}, \quad j = 0, \cdots, P_i - 1 \tag{16}$$

$$y_{ij} \cdot V_{dd} \le V^i_{Src} + I^{ij}_{Src} \cdot \frac{1}{2^j G_z} \le V_{dd}, \quad j = 0, \cdots, P_i - 1 \tag{17}$$

$\forall i \in C4_{fixed}$:

$$y_{i0} = y^{fixed}_{i0} \tag{18}$$

The objective function, as described in Section V-A, represents the number of C4s used, and our goal is to minimize this number.

The constraints can be described as follows:

- In Constraints (13), we use Equation (10) to represent the voltage at the OBS nodes in the coarse grids, and state that this must exceed V^k_{th}, the minimum required voltage at each observation node. At the coarse level $r \ge 1$, we set V^k_{th} to be slightly less stringent than V_{thre}, the specified minimum voltage for any node in the original/flat power grid ($r = 0$), to allow for possible errors in coarsening.
- Constraints (14) ensure that the number of inserted C4s does not exceed l_i, the total number of C4s available at C4 port i.
- Constraints (15) create a lumped current model for the C4 pads in the coarsened grids, and come from Equation (11). In Fig. 5, each Vdd node has P_i switches that determine the resistance of the lumped C4s between Vdd and port i. The term I^{ij}_{Src} represents the current flowing through the j^{th} switch, so that the total current supplied by the supply is the summation of all these terms.
- The second inequality in constraints (16) ensures that the EM constraint is met when this C4 bump is used, i.e., $y_{ij} = 1$. Here, $I^i_{max}(T_i)$ is the maximum allowable current through each C4 bump at port i when its temperature is T_i. Together with the first inequality, this constraint is structured to ensure that the current I^{ij}_{src} through the j^{th} switch is zero when no C4 bump is connected at the j^{th} conductor at C4 port i, since both sides of the inequality are zero.
- Constraints (17) set the bound for the Vdd supply, with the $I^{ij}_{Src}/(2^j G_z)$ term representing the IR drop from Vdd to port i along the path through switch j.
- Constraints (18) fix the C4s in the finished partitions during MILP optimization. Here, y^{fixed}_{i0}s are the solutions to the C4s in the finished partitions. Note that for the finished partitions where the C4s are determined, i.e., $i \in C4_{fixed}$, the y_{ij} values are known and are replaced by constants in the formulation.

The above formulation has a linear objective function, linear constraints, 0–1 integer variables y_{ij}s as well as continuous variables, and is therefore a 0–1 MILP.

D. Complexity Analysis

As presented in Section II, our C4 placement algorithm is based on divide-and-conquer principle, and we solve a series of small MILP problems (Section V-C) to optimize the C4 placement, one for each active partition. Therefore, the runtime complexity of our approach depends on: 1) the size of MILP problem for each active partition, 2) the total number of active partitions in the chip (i.e., leaf nodes in the corresponding quad-tree).

Each MILP formulation in Section V-C has $\sum_i^m P_i$ 0-1 integer variables y_{ij}s, but only $(\sum_i^m P_i - |C4_{fixed}|)$ of which are *free* variables because we fix the C4s in the already solved finished partitions. In addition, it has $(m + \sum_i^m P_i)$ continuous variables (V^i_{Src}s and I^{ij}_{Src}s), $n + 5m$ inequality constraints and $m + |C4_{fixed}|$ equality constraints. The runtime to solve an MILP is largely determined by its number of integer variables (i.e., number of C4 candidates).

In our approach, after breaking the C4 placement problem into small MILPs based on the divide-and-conquer principle, we can solve each of them efficiently. For a given chip, we can also explore the depth of the corresponding quad-tree to find a good partitioning, in order to obtain a reasonable number of active partitions, as discussed in Section II-A.

VI. Experimental Results

Our C4 placement approach described in Section II is implemented in C++. The MILP problems are solved using CPLEX [21]. Table I shows the values of the parameters in the EM model (Section IV) used in our experiments, most of them are taken from [4]. We use HotSpot [22] to obtain the temperature map for the circuit, which is required for the calculation of the maximum current a C4 can carry at each candidate location.

EM limit $MTTF_{min}$	8.76e4 h
Constant A	4.38e4
Current crowding factor c	10
Model parameter n	1.8
Area of C4 bump S	2.5e-5 cm^2
Activity energy Q	0.8 eV
Boltzmann's constant k	8.617e-5 eV/K
C4 resistance R	0.25 Ω
C4 thermal resistance R_θ	40 K/W

TABLE I: Parameters in the EM model.

We use five benchmarks in our experiments which are transformed from the IBM power grid benchmarks [23], using the procedure described in Section III. The VDD is 1.8V for all these circuits and we set the minimum voltage $V_{thre} = 1.6V$ for all the circuits. Columns 2 to 4 of Table II show the size of these circuits, the numbers of C4 candidates and OBS nodes we chose for each circuit.

As stated earlier, our C4 placement approach is based on the divide-and-conquer principle, and a series of small MILP problems are solved instead of solving a large MILP for the whole circuit. We compare our approach with two other MILP-based approaches presented in [2]:

- *Single MILP*: A single MILP is formulated for the whole circuit.
- *Isolated MILP*: Divide and conquer approach is also used to partition the whole circuit, as presented in Section II. But each MILP subproblem is only for the circuit in the current active partition, the circuit in all the other partitions is ignored. In contrast, in our work, we add all the other partitions to the MILP subproblem by including their coarse grid with lumped C4 candidates and OBS nodes.

We first investigate the efficiency of our new EM model described in Section IV. We pick a circuit ckt5, use Hotspot to generate its temperature map, and then obtain the I_{max} map, the distribution of the maximum current a C4 can carry across the chip, for the chip based on Equation (3) and the values listed in Table I. Results show that the I_{max} varies from 1.02A to 1.73A across the chip, the variance in I_{max} is more than 50%. Then we run two different optimizations using the same approach *Isolated MILP*. First, we use the accurate I_{max} map for the C4 optimization, the reported minimum number of used C4s is 198. Next we use a constant $I_{max} = 1.02$A (corresponding to the maximum temperature in the chip) for all the C4 candidates, reflecting the old EM model used in [2], and the reported minimum number of used C4 is 251, which implies more than 25% excessive usage of C4 resource. Therefore, these results show that our model can capture the variation of I_{max} across the chip caused by the variation of temperature, and help planning the optimal layout of C4 pads for a given circuit.

Next we compare the efficiency of three aforementioned MILP-based C4 placement approaches. Columns 5-14 in Table II show the results of these three approaches. For each approach we report *#integer*, the size of the MILP(s) in terms of the number of *free* integer variables, *#used-C4*, the total number of used C4s, and *CPU*,

TABLE II: Comparison of three C4 placement approaches.

Ckt	#nodes	#C4	#OBS	Single MILP			#MILP	Our work			Isolated MILP		
				#integer	#used-C4	CPU		#integer	#used-C4	CPU	#integer	#used-C4	CPU
ckt1	25,921	342	225	342	20	935.6	16	[31,91]	23	34.0	[7,35]	61	0.6
ckt2	141,151	486	380	486	41	1328.4	64	[23,79]	44	41.5	[4,38]	80	1.7
ckt3	640,380	324	782	324	67	963.1	64	[20,47]	72	30.0	[2,20]	99	3.7
ckt4	450,241	656	529	656	60	2118.3	91	[27,85]	66	43.3	[4,26]	124	4.8
ckt5	625,681	799	729	799	96	3037.4	118	[30,84]	103	46.5	[3,38]	198	5.3
Avg					0.50	721			0.54	21		1	1

the total runtime to solve the C4 placement problem (including setup, macromodeling and MILP optimization) in *minutes*. For our approach and *Isolated MILP*, we also list *#MILP* (see Column 8), the number of MILPs solved by each approach, which equals to the total number of active partitions in the chip. All the results are measured on a 64-bit, 2.5GHz Intel Quad-core machine running Linux.

As expected, since *Single MILP* approach solves the C4 placement problem with one-time MILP optimization, it can search for the optimal solution for the whole circuit, and therefore find the best solution. However, the cost to solve a large MILP (see Column 7) is prohibitive – it takes tens of hours to solve the C4 placement problems with hundreds of C4 candidates. Therefore, this approach is inefficient and unscalable. *Isolated MILP* runs fastest, because it only solves a small MILP for each pending partition (see Column 12). However, we see a large overhead in the number of used C4s because this approach does not consider the fact that the C4s used in one region can be shared by the adjacent regions. By contrast, our approach based on divide-and-conquer and grid-coarsening can find good solutions close to those of *Single MILP*, but with significant speedup. For the large circuit ckt5 with about 800 C4 candidates, our approach can find a good solution in less than one hour. In fact, our approach is also scalable – by adaptively adjusting the depth of the partitioning of the chip, our approach provides a scalable solution to large C4 placement problems by controlling the complexity of each MILP subproblem (See Column 9).

VII. Conclusion

In this paper, we present an efficient algorithm for the placement of power supply pads in flip-chip packaging. We use a divide-and-conquer approach to partition the whole circuit into small partitions and then solve small MILP problems for each partition by modeling the distant nodes and sources coarsely following the coarsening phase in multigrid approach. We present a solution to model the lumped C4s with varying conductance in the MILP formulation. We also develop an accurate temperature-dependent EM model for the C4 power pads and integrate this model with our C4 placement algorithm. The effectiveness of the proposed approach is verified on several benchmarks adapted from real designs.

Acknowledgment

This work was supported in part by NSF ECCS-0903427 and SRC 2009-TJ-1990. We would like to thank professor Zhuo Feng at Michigan Technological University for providing the multigrid analyzer. We also owe thanks to all anonymous reviewers for their constructive comments that help improve the quality of this paper.

References

[1] K. DeHaven and J. Dietz, "Controlled collapse chip connection (C4)-an enabling technology," in *Proceedings of Electronic Components and Technology Conference*, 1994, pp. 1–6.

[2] M. Zhao *et al.*, "Optimal placement of power supply pads and pins," in *Proceedings of the ACM/IEEE Design Automation Conference*, 2004, pp. 165–170.

[3] Y. C. Chan and D. Yang, "Failure mechanisms of solder interconnects under current stressing in advanced electronic packages," *Progress in Materials Science*, vol. 55, no. 5, pp. 428–475, July 2010.

[4] W. J. Choi *et al.*, "Mean-time-to-failure study of flip chip solder joints on Cu/Ni(V)/Al thin-film under-bump-metallization," *Journal of Applied Physics*, vol. 94, no. 9, pp. 56–65, 2003.

[5] D. S. Chau *et al.*, "Experimental method of measuring C4 die bump temperature for electronics packaging," in *the Ninth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2004, pp. 91–95.

[6] K. N. Chiang *et al.*, "Current crowding-induced electromigration in $SnAg_{3.0}Cu_{0.5}$ microbumps," *Applied Physics Letters*, vol. 88, no. 7, Feb. 2006.

[7] C. Chen *et al.*, "Electromigration and thermomigration in Pb-free flip-chip solder joints," *Annual Review of Materials Research*, vol. 40, pp. 531–555, Aug. 2010.

[8] J. Oh and M. Pedram, "Multi-pad power/ground network design for uniform distribution of ground bounce," in *Proceedings of the ACM/IEEE Design Automation Conference*, 1998, pp. 287–290.

[9] T. Sato *et al.*, "Successive pad assignment algorithm to optimize number and location of power supply pad using incremental matrix inversion," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2005, pp. 723–728.

[10] Y. Zhong and M. D. F. Wong, "Fast placement optimization of power supply pads," in *Proceedings of the Asia and South Pacific Design Automation Conference*, 2007, pp. 763–767.

[11] P. Zhou *et al.*, "Optimization of on-chip switched-capacitor DC-DC converters for high-performance applications," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2012, pp. 263–270.

[12] M. Zhao *et al.*, "Hierarchical analysis of power distribution networks," in *Proceedings of the ACM/IEEE Design Automation Conference*, 2000, pp. 150–155.

[13] M. R. Bussieck and A. Pruessner, "Mixed-integer nonlinear programming," *SIAG/OPT Newsletter: Views & News*, 2003.

[14] E. Chiprout, "Fast flip-chip power grid analysis via locality and grid shells," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2004, pp. 485–488.

[15] J. Singh and S. S. Sapatnekar, "A fast algorithm for power grid design," in *Proceedings of the International Symposium on Physical Design*, 2005, pp. 70–77.

[16] J. N. Kozhaya *et al.*, "A multigrid-like technique for power grid analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 10, pp. 1148–1160, Oct. 2002.

[17] K. Wang and M. Marek-Sadowska, "On-chip power-supply network optimization using multigrid-based technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 407–417, Mar. 2005.

[18] Z. Feng *et al.*, "Parallel on-chip power distribution network analysis on multi-core-multi-gpu platforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1823–1836, Oct. 2011.

[19] K. Banerjee and A. Mehrotra, "Global (interconnect) warming," *IEEE Circuits and Devices Magazine*, vol. 17, no. 5, pp. 16–32, Sep. 2001.

[20] C.-W. Ho *et al.*, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[21] "IBM ILOG CPLEX Optimization Studio v.12," available at http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

[22] W. Huang *et al.*, "HotSpot: a compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[23] "IBM power grid benchmarks," available at http://dropzone.tamu.edu/~pli/PGBench/.