UNIVERSITY OF CALIFORNIA
SANTA CRUZ

THERMAL-AWARE CAD FOR MODERN INTEGRATED CIRCUITS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Sheldon Logan

June 2013

The Dissertation of Sheldon Logan
is approved:

Professor Matthew R. Guthaus, Chair

Professor Jose Renau

Professor Martine Schlag

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Thermal-Aware CAD for modern integrated circuits

by

Sheldon Logan

Power density in modern integrated circuits (ICs) continues to increase at an alarming rate. In turn, larger power densities result in higher peak temperatures which can reduce chip reliability and further increase leakage power consumption. Thermal-aware CAD design is a method to combat these problems. However most existing thermal-aware CAD research has focused on thermal-aware floorplanning and placement. These thermal-aware floorplanners have several problems such as long execution times and being limited to only one method of reducing peak temperatures. In addition most thermal-aware CAD research has only focused on reducing chip peak temperatures and not other reliability concerns such as high interconnect temperatures, wire/C4 bump electromigration, and thermal-cyclic C4 bump failure.

This thesis proposes several new algorithms and methodologies that can be used to directly reduce high on chip temperatures and mitigate the reliability concerns caused by these high temperatures. Experimental results show that the proposed thermal-floorplanning moves based on whitespace utilization, coupled with a method of quickly evaluating temperature effects can reduce on chip-temperatures on average by 7K with only a modest 4.2% increase in wirelength and 1.12x increase in execution time. In addition, a method for co-optimizing floorplanning and C4 bump placement using a quadratic optimization process is shown to increase the lifetime of bumps from thermal-cyclic fatigue by $47\times$ with only

a modest 3% increase in HPWL wirelength. To combat bump electromigration, a single-bump redundancy technique based on Integer Linear Programming (ILP) is proposed, and shown to be able to reduce the number of redundant bumps to guarantee single-bump redundancy by 68% as compared to a naive bump placement approach. Finally, an algorithm to redistribute decoupling capacitance, is shown to be able to reduce interconnect temperatures by 12.5K on average and provide a $1.66\times$ increase in electromigration lifetime.

Acknowledgments

First of all, I would like to thank God, my Lord and saviour for guiding me through the many troubles and tribulations that I have encountered through my PhD.

I would like to also express my deepest gratitude to my adviser Matthew Guthaus, without his help and motivation I would have never seen the finish line. I thank him for initially taking me into his lab during my second year in graduate school at a point where I was lost and indifferent about graduate school and research. I thank him for his initial patience as I got up to speed in CAD and his continual motivation along the way, which quelled the many thoughts I have had of quitting. More importantly, I want to thank him for his contribution of ideas, thoughts and funding, which allowed me to finish my PhD.

In addition to my adviser, I would like to thank the rest of my thesis committee: Professor Jose Renau and Professor Martine Schlag for their assistance for helping me complete my thesis.

I thank my fellow lab-mates in UCSC VLSI-DA group: Seokjoong Kim, Xuchu Hu, Rajsaktish Sankaranarayanan, Marcelo Siero, Walter Condley, Derek Chan and Keven Woo, for their friendships but more importantly the ideas and contributions that allowed me to finish my PhD.

I also wish to thank the graduate adviser Carol Mullane, for her help during my PhD journey. She always willing to provide suggestions and helped me with a lot of paper-work and special problems.

I would also to like to thank my friends from church and from graduate school (Audries Blake, Emily Scheese, Sarah Romano, Neil Miller and Mikhail Rudenko) that I

have accumulated during my time in Santa Cruz who encouraged me to reach my goal.

Last but not the least, I would thank my parents and extended family that helped me along the journey with their countless words of motivation and support. Without them, it would have been impossible to complete my journey.

Chapter 1

Introduction

High on-chip temperatures have quickly become one of the major concerns for modern integrated circuit (IC) designers. Extreme power densities due to the aggressive scaling of transistor sizes have resulted in large peak temperatures and drastic temperature gradients. These large peak temperatures and gradients lead to several reliability concerns in modern integrated circuits such as electromigration, NBTI, increases in leakage power and thermal-cyclic fatigue. Consequently, most designers now consider temperature, along with power, in the early parts of the design phase.

Reducing the overall chip temperature is one of the best methods of combating the reliability concerns caused by high temperatures in modern ICs and this is usually done via thermal-aware floorplanning [7,17,18,26,33,48,49,62,114]. Thermal-aware floorplanning, consists of adjusting the cost function of a floorplanner to include some temperature metric in addition to the other typical design metrics.

Thermal-aware floorplanning however, can only reduce the high IC temperatures by a certain amount with the resulting IC temperatures potentially still leading to other

reliability problems such as mechanical failure of C4 solder bumps. These bumps fail due to thermal-cyclic fatigue which is caused by the coefficient of thermal expansion (CTE) mismatch between the substrate and package.

Another reliability issues caused by high IC temperatures is the increase in interconnect resistivity and electromigration. Resistivity increases result in IR drop and voltage droop violations leading to timing and signal integrity issues [88]. Increases in interconnect electromigration can potentially lead to breaks in the interconnect or failure of C4 bumps, which would result in complete IC failure.

The increases in resistivity and electromigration are only going to worsen as designs move to smaller technologies. Decreased wire cross section areas, lower supply voltages and low-K dielectrics between metal layers all increase interconnect temperatures [12, 28, 36]. Lower supply voltages result in higher currents due to the inverse proportional relationship between voltage and current. Low-K dielectrics exacerbate the temperature problems since the dielectric used to electrically insulate the various metal layers have very low thermal conductivity which leads to poor dissipation of heat from the metal wires, especially in the higher layers. 3D-ICs further suffer from this problem because interconnect layers in the highest tiers are located very far from the heat sink.

Other reliability issues caused by high IC temperatures are increases in NBTI and increases in circuit leakage. NBTI is the degradation of the threshold voltages, drive currents and noise margins in negative bias transistors. These degradations can lead to circuit timing errors and in the worst case, catastrophic circuit failure. The magnitude of circuit current leakage is strongly correlated to temperature and thus, high chip temperatures can lead to dramatic increases in power usage due to the sudden increase in leakage current.

1.1 Thesis Contributions and Outline

This thesis proposes several CAD algorithms and methodologies that reduce high IC temperatures and also mitigate the reliability issues caused by these temperatures. Chapter 2 introduces the background for the algorithms and methodologies and Chapter 3 introduces the thermal background knowledge required to understand them. The algorithms and methodologies are detailed in Chapters 4, 5, 6, 7. The experimental setup for the algorithms and methodologies are presented in Chapter 8 and the results and discussions from these experiments are detailed in Chapter 9. Finally the contributions of this thesis are concluded in Chapter 10.

Chapter 4 contains the contributions to thermal floorplanning. It introduces an effective way to lower maximum temperatures in floorplanning by adjusting white space usage, in addition to moving blocks around in a floorplan, which is what is typical done. It also presents a power metric cost for thermal floorplanning that is faster than other methods and still results in excellent solutions [53].

Chapter 5 contains the contributions to thermal-cyclic bump fatigue. It first introduces a simplified stress/strain/fatigue model for C4 solder bumps that can be used during floorplanning to guide C4 bump placement. It then presents a quadratic C4 bump placement algorithm that optimizes for both wirelength and reliability which can be used to increase the lifetime of C4 bumps from thermal-cyclic fatigue [54].

Chapter 6 contains the contributions to the area of reducing interconnect temperatures. It presents a methodology for the redistribution and/or allocation of decoupling capacitance (decap) to reduce the Joule heating power in the PSN interconnect. The method-

ology is based on a gradient-based algorithm and can lead to significant increases in PSN integrity.

Finally Chapter 7 contains the contribution to reducing the impact of interconnection electromigration in circuits. It presents a methodology to create PSNs that are single-bump redundant, meaning if any bump fails due to electromigration and/or manufacturing the PSN will still be able to meet its' requirements. The methodology uses an Integer Linear Program (ILP) to generate the smallest set of redundant bumps to guarantee single-bump redundancy. The chapter also presents a methodology which uses thermal modeling for determining which bumps are critical to the PSN design and it is also the first work to consider C4 bump manufacturing defects in PSN design.

Chapter 2

Background and Previous Work

This chapter provides background information on the various VLSI problems addressed throughout the thesis and also details the CAD algorithms and methodologies proposed by other researchers to combat these problems. Sections 2.1, 2.2, 2.3 and 2.4 analyzes the floorplanning problem and shows how previous researchers have tackled thermal-aware floorplanning. Sections 2.5 introduces the concept of flip-chip packaging and analyzes the flip-chip C4 bump placement problem. Finally, Sections 2.6, 2.7 and 2.8 introduces the concept of a Power Supply Network (PSN) and then analyzes the various problems associated with PSN synthesis: wire sizing, decap placement, and power bump placement.

2.1 Floorplanning

The floorplanning problem is defined as follows: Given a set of blocks, find the optimal orientation of those blocks to minimize certain design metrics such that no blocks overlap. The classical floorplanning design metrics are area and wirelength. Recently floorplanning has been extended to consider other metrics such as maximum temper-

ature [7, 17, 18, 26, 33, 48, 49, 62], power supply noise [10, 111] and leakage [27, 114]. This thesis focuses on the classical metrics in addition to maximum temperature.

2.1.1 Area

The floorplan area metric (Area_f) is calculated as

$$\mathrm{Area}_f = \mathrm{width}_{\max} \times \mathrm{height}_{\max} \tag{2.1}$$

where $\mathrm{width}_{\max}$ is the maximum width of the floorplan and $\mathrm{height}_{\max}$ is the maximum height of the floorplan. If the area of the floorplan is larger than the total area of the blocks then the floorplan contains whitespace. More formally, floorplanning whitespace is defined as any area within the floorplan that is not being occupied by a block. Figure 2.1 shows two floorplans for a circuit. Figure 2.1(a) contains whitespace while Figure 2.1(b) contains no whitespace.

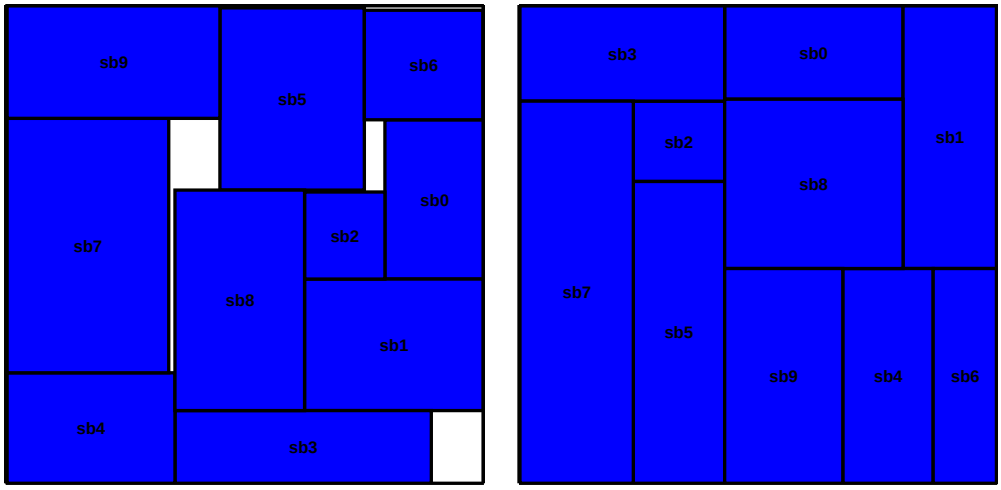
Recently, researchers have focused on fixed-area floorplanning [1, 3]. In fixed-area floorplanning the area metric is modified to

$$\mathrm{Area}_f = \begin{cases} 1 & \text{if } (\mathrm{width}_{\max} \times \mathrm{height}_{\max} < \mathrm{Area}_{fix}) \\ \frac{\mathrm{width}_{\max} \times \mathrm{height}_{\max}}{\mathrm{Area}_{fix}} & \text{otherwise} \end{cases} \tag{2.2}$$

where Area_{fix} is the area designated to hold the floorplan.

2.1.2 Wirelength

The wirelength metric is used to estimate the total wiring required to route the pins within the floorplan. There are three main methods to estimate wirelength, Half Perimeter Wirelength (HPWL), Steiner Wirelength and rectilinear Minimum Spanning Tree



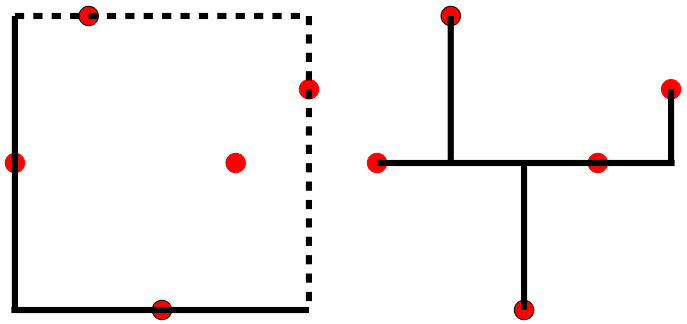
(a) Floorplan with whitespace Area_f > Total area of blocks

(b) Floorplan with no whitespace. Area_f = Total area of blocks

Figure 2.1: Two floorplans for the same circuit illustrating the concept of floorplanning whitespace

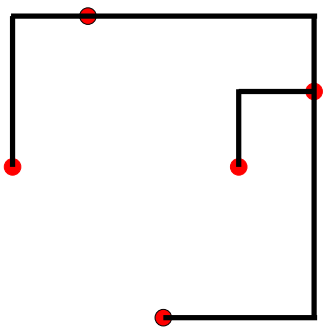
(MST) Wirelength as shown in Figure 2.2. HPWL is the estimate used in this thesis since most floorplanners in literature use this metric [1,2,7,8,18,33,40,49,62,114] and also due to the ease and quickness of HPWL calculation. Some floorplanners in the past however have used the other estimation techniques [73,79].

HPWL estimates wirelength by creating a bounding box around all the pins and taking half the perimeter of the bounding box as an estimate of the total wiring required to route that pin as shown in Figure 2.2(a). Steiner based estimates calculates the wirelength by creating a Steiner tree using the pins as vertices and using the sum of the edge lengths of the tree as the wirelength estimate as shown in Figure 2.2(b). Minimum spanning tree estimates calculates the wirelength by creating a Minimum Spanning tree using the pins as vertices and using the sum of the edge lengths of the tree as the wirelength estimate as



(a) Routing estimate for 5 pins using the HPWL estimate

(b) Routing estimate for 5 pins using the Steiner wirelength estimate



(c) Routing estimate for 5 pins using the MST estimate

Figure 2.2: The differences in routing estimates for 5 pins using the various wirelength metrics.

shown in Figure 2.2(c).

Wirelength estimation is also used during floorplanning to estimate routing congestion within floorplanning blocks. Blocks with potential high routing congestion have their area increased as a method of mitigating potential routing problems. This technique is

called area utilization.

2.1.3 Temperature

The temperature metric of choice for most floorplanners is the maximum temperature of the floorplan [7,17,18,26,33,48,49,62,114]. However, using maximum temperature as a floorplanning metric significantly increases the runtime of the floorplanner since computing thermal profiles takes much longer than calculating the other metrics such as floorplan area or HPWL. Most previous floorplanning research has used grid-based thermal simulators [26, 33, 114] which require significant amount of time to compute thermal profiles. Some researchers [17,48] have tried to decrease the runtime for calculating thermal profiles by using a simplified thermal model, but this results in significant loss of accuracy in maximum temperature calculations and consequently suboptimal floorplanning solutions. In general, most floorplanners trade off accuracy in temperature calculations for run-time. The temperature is typically modeled using detailed finite-element (FEM) simulation [11, 96], compact resistive network modeling [82], or other approximations [67, 94, 107].

2.2 Floorplanning Representation

Each floorplan in the solution space is represented using some form of floorplanning representation such as Sequence Pair (SP) [58], B*-tree, [8], Corner Block List [31, 56], slicing tree [43, 105] or Transitive Closure Graph TCG [50]. The floorplanners proposed in this thesis are based on SP representation due to its simple data structure and efficient representation.

2.2.1 SP Representation

The SP representation consists of two sequences, positive and negative step lines, which capture the geometric information of a floorplan. Figure 2.4 and Figure 2.5 show the positive and negative step lines respectively for the floorplan shown in Figure 2.3. The sequence pair for the floorplan is thus <4 1 2 3> (positive) and <1 2 4 3> (negative).

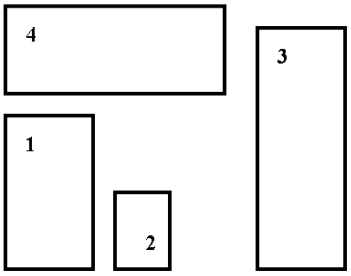


Figure 2.3: A Floorplan Containing 4 Blocks showing the Geometric Relationships between the Blocks

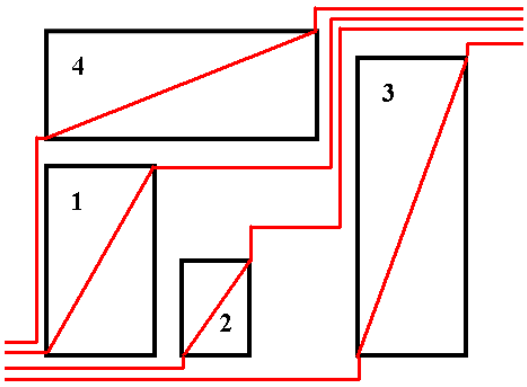


Figure 2.4: The Ordering of Blocks based on Positive Step Lines <4 1 2 3>

The order of the blocks in the two sequence determines the geometric orientation of the blocks. If block x occurs before block y in both sequences then block x is to the left

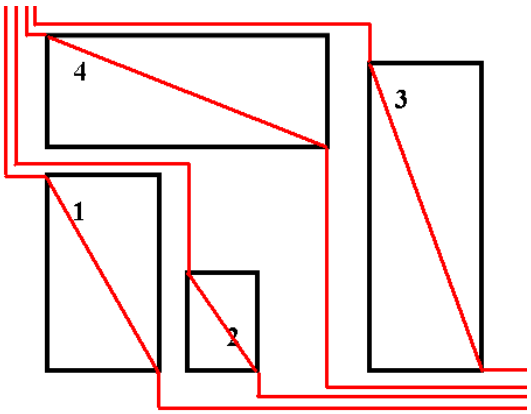


Figure 2.5: The Ordering of Blocks based on Negative Step Lines <1 2 4 3>

of y. An example of this relation is blocks 1 and 3. However, if block x occurs before block y in the positive sequence and block y occurs before block x in the negative sequence then block x is above y. Blocks 1 and 4 are a good example of this relation. There are several methods of obtaining a floorplan from a sequence pair. The method that is used in the thesis is longest common subsequence ([86]).

Another method of compacting a sequence pair uses horizontal and vertical constraint graphs. These graphs are constructed from the "left of" and "below" constraints obtained from the sequence pair. An example of the horizontal and vertical constraint graphs obtain from the floorplan in Figure 2.3 are shown in Figure 2.6 and Figure 2.7, respectively. The weight of the vertices in the horizontal and vertical constraint graphs are the width and height of the blocks. The longest path in the vertical and horizontal constraint graph represents the height and width of the chip. The x coordinate and y coordinate of a block is calculated by finding the longest path to that block in the horizontal and the vertical constraint graphs respectively. Consequently the blocks are packed in a down and to the left

manner.

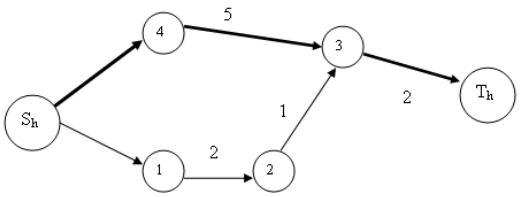


Figure 2.6: Horizontal Constraint Graph

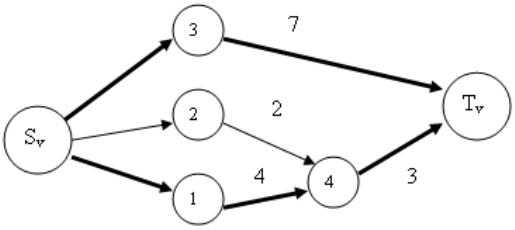


Figure 2.7: Vertical Constraint Graph

2.3 Floorplanning Methods

There are many different methods of solving the floorplanning problem. Most of the methods however can be separated into two categories, iterative approaches and constructive approaches. Iterative methods include simulated annealing [1, 2, 8, 17], force-directed approaches [114] and genetic algorithms [33]. Constructive methods include analytical [40] and structural partitioning [16]. The method of floorplanning used throughout the thesis is simulated annealing since it is the method used by most previous thermal-aware floorplanners [7, 15, 17, 33, 102] and its ease of implementation.

2.3.1 Genetic Algorithms

Genetic Algorithms are iterative methods of searching through a solution space that try to mimic evolutionary principles. They make changes to an initial set of solutions (called an initial population) by the use of three operators: crossover, mutation and reproduction. Each initial solution is called a chromosome and the quality of each solution is evaluated using a fitness function. Reproduction finds high quality chromosomes and duplicates them. Crossover selects two random chromosomes and swaps some portions of each chromosome with some probability. Finally, the mutation operator takes a chromosome and randomly changes a portion of it. These three operators are applied to the initial population to produce a new population and the process is repeated until a stopping criterion is met. Hung *et al.* [33] proposed a thermal-aware floorplanner based on genetic algorithms. It uses a slicing-tree floorplan representation and a grid-based thermal simulator [82]. The major disadvantage of this method of thermal-aware floorplanning is long execution times required by the thermal simulator.

2.3.2 Force Directed

Force Directed techniques for floorplanning transforms the problem to a statics problem of force balancing. The connectivity between blocks acts as attractive forces while filling forces are introduced to spread apart blocks, remove overlap and reduce floorplanning whitespace. Zhou *et al.* [114] proposed a thermal-aware floorplanner using a force-directed approach. In addition to the normal filling forces and attractive forces the authors incorporate a thermal force which is used to move hot blocks away from areas of high temperature.

One of the major disadvantages of using force-directed floorplanning methods is dealing with overlap. Usually after all the force balancing has been completed there is significant overlap hence there has to be a post processing step to remove any remaining overlap which will affect the quality of the solution. The second disadvantage of this method is runtime required to calculate the temperatures needed for the thermal forces.

2.3.3 Simulated Annealing

Simulated Annealing [42] is an iterative method of searching large solution spaces for a close to optimal solution. It consists of making incremental changes to an initial solution and accepting or rejecting the change based on the difference in the optimality between the two solutions. If the new solution is better than the old solution the change will always be accepted. However, to avoid being caught in local minima, if the change is worse than the new solution might be accepted depending on difference in optimality and the elapsed time of the annealer. More bad moves are accepted if the elapsed time is small and/or if the difference in optimality is small. The quality of a solution is evaluated by some cost function.

2.4 SP Simulated Annealing Floorplanning

SP-based floorplanning forms the basis of most previous thermal-aware floorplanners [7, 15, 17, 75, 102] and entails searching through the floorplanning solution space (represented as sequence pairs) using simulated annealing. The search is accomplished by modifying the sequence pair and the blocks in the floorplan. The quality of the floorplan-

ning solution is evaluated using a simple cost function.

2.4.1 SP Floorplanning Moves

Typically, SP floorplanners have four moves. The first move is to swap the location of two blocks by swapping two pair of numbers in both sequence pairs. The second moves entails moving a single block by swapping a pair of numbers in only one sequence pair. The third move rotates a block which entails swapping a blocks height for its width. And the final move changes the aspect ratio of a soft block by adjusting its height and width.

2.4.2 SP Floorplanning Cost Function

The cost function for a SP based floorplanner usually includes the classical floorplanning metrics area and HPWL. Thermal-aware floorplanners [15, 17, 75, 75, 102] include the additional metric maximum temperature, leading to the following cost function

$$cost = \alpha \cdot \text{Area}_f + \beta \cdot \text{HPWL} + \eta T_{Max} \tag{2.3}$$

where T_{Max} is the maximum chip temperature and α, β, η, are the different weights associated with each value.

A major disadvantage of using T_{Max} in a cost function is that only moves that directly affect the hottest temperature hotspot decrease the cost function. If a move significantly decrease temperatures in the other hotspots of the chip the cost function is not decreased and the annealer will more than likely not accept the move decreasing the efficiency of thermal-based floorplanning moves. The second disadvantage of using T_{Max} is the long computation times required to calculate floorplan temperatures compare the other metrics, HPWL and area.

2.5 Flip Chip Packaging

There are two main methods of connecting IC dies to package substrates, wire bonding and flip chip. Wire bonding routes bond wires from the I/O pads located on the perimeter of the die to the package. The flip chip method, also called Controlled Collapse Chip Connection (C4), connects the die to the substrate using C4 solder bumps located throughout the die area. A sideways view of a flip chip package is shown in Figure 2.8 highlighting the C4 solder bump connections between the chip and package.

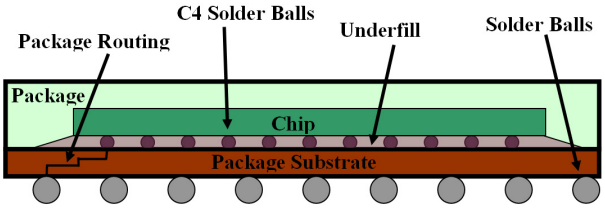


Figure 2.8: Flip Chip Package showing C4 connections.

There are several advantages of flip chip designs over wire-bonded designs. Since the I/O connections can be located throughout the die, the length of wires connecting the die to the package can be greatly reduced leading to better performance. Also, since the I/O connections are not limited to the perimeter of the die, more I/O connections are feasible than in wire bonded designs. Lastly, wire bonded designs usually require larger packages due to the I/O pads and bonding wires. These advantages along with the increase in complexity and size of ICs have resulted in many modern designs using flip chip connections [68].

2.5.1 C4 Bump Placement and Floorplanning Co-optimization

The C4 bump placement problem can be defined as follows: Given a set of required package chip connections find the optimal placement of C4 bumps to minimize certain design metrics. The floorplan of the circuit affects the location of C4 bumps especially for data I/O connections, consequently, most researchers have tried to optimize the location of C4 bumps and the floorplan concurrently. Most previous methods of C4 bump placement/floorplanning have focused on minimizing design metrics such as total wirelength and skew [32, 46, 70, 80, 95]. Other works have focused on how the C4 placement affects the PCB routability [21]. None of these works however consider the C4 bump reliability as a design metric although there have been studies on the reliability of C4 bumps in flip-chip packages [44, 65, 78, 90].

2.6 Power Supply Networks (PSNs)

PSNs are used in ICs to deliver power to the various transistors that comprise the circuit. A PSN contains power bumps from the package, wires (interconnect) and decoupling capacitors (decaps). The power bumps and the decaps supply current to transistors in the circuit. The interconnect distributes the current throughout the circuit from the power bumps/decaps to the transistors. There are several possibilities available for the topology of the wires in the PSN, namely trees and grids. In most modern designs however, grids are mainly used and thus this thesis focuses on power supply grids. The decaps act as a local energy supply and assist in reducing the dynamic voltage droop ($L\frac{di}{dt}$) caused by the sudden current draw of the localized transistor switching.

2.6.1 Electrical Modelling

PSNs are modelled using simple electrical elements such as current sources, voltage sources, resistors, capacitors and inductors. The power bumps are modeled as inductors in series with resistors attached to an ideal, off-chip voltage source. The interconnect is modeled as a network of resistors with some inductance and capacitance. The transistors are modeled as individual, distributed current sources with a small amount of diffusion capacitance. An example of an electrical circuit for a power grid is shown in Figure 2.9. It should also be noted that the current drawn by transistors is macro-modeled to occur at each mesh node for easier calculation. In reality, the number of transistors (current sources) would be much larger than the number of nodes in the mesh.

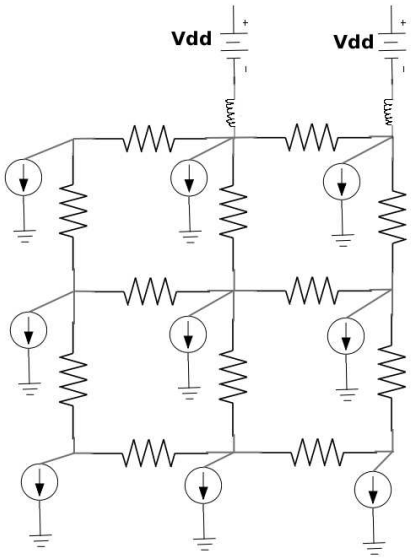


Figure 2.9: A Simple VDD Power Supply Network showing the basic Circuit Elements used to model the PSN.

2.6.2 Electrical Simulation

The voltages for a PSN are calculated using Modified Nodal Analysis (MNA) according to

$$G \cdot v\left(t\right) + C \cdot \dot{v}\left(t\right) = i\left(t\right) \tag{2.4}$$

where G is the conductance matrix, C is the admittance matrix (inductance and capacitance elements), $v(t)$ represents the time varying nodal voltages and $i(t)$ represents the vector of current sources corresponding to the transistors in the design. The conductance and admittance matrices are obtained by applying Kirchhoff's current law at each node of the circuit. If the backward Euler method is used to solve the transient system, Equation 2.4 can be written as:

$$\left(\frac{C}{h} + G\right) v\left(t\right) = i\left(t\right) + \frac{C}{h} v\left(t - h\right) \tag{2.5}$$

where h is the time step of the simulation.

For PSN steady state simulations the admittance matrix is ignored and Equation 2.4 becomes

$$G \cdot v = i \tag{2.6}$$

where v and i are the steady state node voltages and transistor currents of the PSN.

2.7 PSN Requirements

A typical PSN has three requirements: 1) The steady state current density through each power bump and the interconnect should be below a certain threshold to ensure no

failures from electromigration. 2) The DC voltage at each node in the circuit should be above a specified voltage to ensure the correct functionality of the circuit. Lower voltages causes transistors to switch slower and can significantly impact the timing of the circuit. 3) The transient voltage droop caused by the sudden switching of transistors should be above a certain threshold to ensure the correct functionality of the circuit.

2.7.1 Voltage Droop

Voltage Droop is defined as the decrease in voltage in the PSN as the transistors turn on and current is drawn through the interconnect as shown in Figure 2.10.

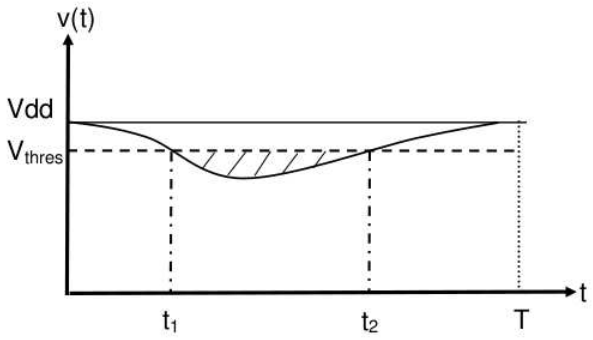


Figure 2.10: Voltage drop at a power supply node caused by the current draw of a transistor.

Occasionally the voltage might droop below the voltage minimum leading to timing issues in circuits. The voltage droop g_j at a specific node j in the PSN is defined to be

$$g_j = \int_0^T \max\left(V_{thres} - v_j\left(t\right), 0\right)\, \mathrm{d}t \tag{2.7}$$

where T is the clock period, V_{thres} is the minimum voltage threshold, and $v_j\left(t\right)$ is the voltage at node j as shown in Figure 2.10.

2.8 PSN Synthesis

PSN synthesis creates a PSN network that meets certain requirements(Section 2.7). It entails sizing interconnect, placing decap and finally placing power supply bumps to supply the PSN with energy.

2.8.1 Wire Sizing

The wire sizing problem can be defined as follows: Given the interconnect network for a PSN, size individual wires optimally within the routing constraint. Larger wires have smaller resistances which results in lower amounts of IR drop and voltage droop. Larger wires also tend to have smaller amounts of electromigration since the amount of electromigration is proportional to the size of the wire. However all the wires within a PSN cannot be made large due to the routing constraints of modern ICs. Routing resources have to be saved for data nets and the clock network.

Researchers have proposed several methods to solve the wire sizing problem. Sapatnekar *et al.* [81] proposed a heuristic which partitions the PSN interconnect recursively and then optimally size the wires in each partition. Wang *et al.* [97] proposed a method of solving the wire sizing problem using a sequential network simplex algorithm. Both of these works, however, size wires for electromigration constraints and IR drop but do not consider other reliability issues such as Joule heating in the interconnect. Recently, researchers have proposed PSN wire-sizing algorithms that optimize for the traditional metrics in addition to Joule heating in the interconnect [99].

Researchers have attempted to solve the joule heating problem in the PSN inter-

connect through other methods than wire sizing. Yokogawa *et al.* . demonstrated that stacking vias between the metal wires and the substrate can help decrease temperatures in PSN interconnect due to increased thermal conductivity between the wires and substrate [104]. Lele *et al.* proposed a method to reduce interconnect Joule heating by optimally sizing signal repeaters [38], but this method is not applicable to PSN interconnects since no repeaters are used.

2.8.2 Power Bump Placement

The power bump placement problem is defined as follows: Place VDD and GND power bumps to ensure that all the PSN requirements are met while minimizing the number of VDD and GND bumps used.

Researchers have proposed many different algorithms to solve the power bump placement problem [37, 64, 77, 109, 113]. Sato *et al.* [77] proposed a greedy formulation that successively places power bumps at the points of highest IR drop in the circuit until the IR drop requirement has been met. To accelerate the algorithm, an incremental matrix inversion is proposed that decreases the computation time required for each PSN simulation. The algorithm does not consider the electromigration constraints for each bump.

Zhong *et al.* [113] proposed a bump placement methodology that is based on simulated annealing. They use an iterative solver to accelerate the computation time for each PSN simulation that is required for each simulated annealing move. The major disadvantage of this approach is that number of bumps is fixed since searching through the solution space with a variable number of bumps is prohibitively time consuming using simulated annealing. This approach also does not consider electromigration in the power bumps

Zhao *et al.* [109] proposed a Mixed Integer Linear Program (MILP) formulation that places bumps so as to satisfy electromigration and IR drop constraints. The MILP formulation proposed is

$$\begin{aligned}
\min \quad & \textstyle\sum_{i \in PC} z_i,\ z_i \in \{0,1\} \\
\text{s.t.} \quad & I_{thres} \cdot z_i - I_i \geq 0 \\
& I_i \geq 0 \\
& v_i - \text{VDD} \cdot z_i \geq 0 \\
& v_i \geq V_{thres} \\
& v_i \leq \text{VDD} \\
& V \text{and } I \text{ satisfy Equation 2.6}
\end{aligned}$$

where PC is a set of all possible bump locations, z_i is a binary variable indicating whether a bump is located at bump position i, I_i is the current through a power bump i, v_i is the voltage of node i, V is the set of all node voltages, I is the set of all node currents, I_{thres} is the maximum allowable current through a power bump, V_{thres} is the maximum tolerable voltage drop in the circuit and VDD is the voltage of a power bump. One major disadvantage of this method is the significant runtime required to solve MILP's that contain many variables. The authors propose methods of macro-modelling the PSN to reduce the problem size which unfortunately leads to solutions that can be far from optimal.

All the aforementioned works on power bump placement have focused on creating an initial bump placement that minimizes the total number of power bumps while ensuring power supply integrity including static voltage violations and electromigration failure of power bumps. They do not consider the robustness of the bump placement in the presence of a single bump failure.

2.8.3 Decap Placement

The decap placement problem is defined as follows: Place the minimum amount of decap so that the transient voltage droop is above a specified threshold. More formally the decap insertion problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} g_j\left(c_1, \ldots, c_m\right) \\ \text{subject to} \quad & c_i \leq d_i \text{ and } \sum_{j=1}^{m} c_i \leq C_{tot} \end{aligned} \tag{2.8}$$

where c_i is the capacitance at node i, d_i is the maximum capacitance at node i and C_{tot} is the maximum total capacitance. g_j is the voltage droop at node j as previously shown in Equation 2.7.

Researchers have proposed several methods for doing decap placement [23,39,47, 52,55,76,84,101,103,110–112]. These methods can be separated into two main categories, those that are based on sensitivity analysis [23, 47, 71, 84] and those that try to estimate the decap to be inserted at a node based on the magnitude of the voltage droop [101, 110–112]. Other methods of solving the decap allocation problem have been proposed in literature. Some researchers have formulated the decap placement problem as a semidefinite program [39, 52]. Other researchers have focused on the decap placement problem in 3D ICs [76, 115].

The sensitivity used by most researchers compute the change in total voltage droop with respect to capacitance and is formally defined as follows [23, 47, 71, 84]:

$$s_{ij} = \frac{\partial g_j\left(c_1, \ldots, c_m\right)}{\partial c_i} \tag{2.9}$$

where s_{ij} is the sensitivity of decap allocated at node i to removing power supply drop g_j at node j. These sensitivities are calculated using adjoint sensitivity analysis. The sensitivity

$s_{i,all}$ of a decap allocated at node i to remove power supply at all nodes can be found using

$$s_{i,all} = \int_0^T \left(v'_{i,all}\left(T-t\right)\right) \times \dot{v_i}\left(t\right)\, \mathrm{d}t \tag{2.10}$$

where $\dot{v_i}\left(t\right)$ is the derivative of the voltage at node i and $v'_{i,all}\left(T-t\right)$ is the voltage at node i in the adjoint network.

The decaps for each node are then calculated using a conjugate gradient method once the sensitives shown in Equation 2.10 are computed [23, 47, 84]. Another method of solving Equation 2.8 given the sensitivities calculated in Equation 2.10 is a sequence of linear programs [71]. In this approach the non linear equations governing the relationship between the sensitivity, droop and capacitance are linearized and then repeatedly solved using a linear programming solver until the total droop has been minimized or the decoupling capacitance budget has been exhausted. The sequence of linear programming optimization can be formally defined as

$$\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{m} c_j & & \\
\text{subject to} \quad & g_j & \leq & \sum_{j=i}^{m} s_{ij}\Delta c_i \\
& c_i & \leq & d_i \\
& \sum_{j=1}^{m} c_j & \leq & C_{tot}
\end{aligned}$$

One disadvantage of using this approach is that the runtime of the algorithm is dependent on the number of violation nodes in the design. Calculating the individual sensitivities s_{ij} for each node requires building and solving an the adjoint network specific to that node, consequently a design with a large number of violation nodes will require many adjoint simulations and will have a significant runtime.

Chapter 3

Thermal Background

This chapter provides the thermal background information needed to understand the proposed thermal-aware CAD algorithms and methodologies detailed in the thesis and is separated into three parts. Sections 3.1 and 3.2 give a brief overview of heat generation and substrate temperature modeling respectively. Section 3.3 introduces models used for the various thermal reliability concerns

3.1 Heat Generation

There are two main sources of heat generation in modern ICs. The majority of the heat is produced from transistors while the rest is produced as Joule heating in the IC interconnect.

3.1.1 Transistor Switching Heating

Most heat energy in ICs is created by the transistors. The total power consumed by a transistor is separated into three major components: dynamic, short-circuit and leak-

age. The total is

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{short-circuit}} + P_{\text{leakage}} \tag{3.1}$$

where P_{total} is the total transistor power, P_{dynamic} is the transistor dynamic power, $P_{\text{short-circuit}}$ is the short circuit power and P_{leakage} is the leakage power.

The dynamic power component depends on several factors such as the switching activity of the transistor, the clock frequency and the load capacitance. More formally the dynamic power of a transistor is

$$P_{\text{dynamic}} = 0.5\alpha_s \text{V}_{\text{dd}}^2 f_c C_L \tag{3.2}$$

where α_s is the transistor activity factor (switching probability), V_{dd} is the supply voltage value, f_c is the clock frequency and C_L is the load capacitance [59].

The short circuit power for a transistor is

$$P_{\text{short-circuit}} = I_{\text{short-circuit}} \text{V}_{\text{dd}} \tag{3.3}$$

where $I_{\text{short-circuit}}$ is the short circuit current which depends on several factors such as α_s, C_L, threshold voltage, signal slew, carrier mobility, f_c and V_{dd} [5]. Both the threshold voltage and carrier mobility are dependent on temperature, consequently, increasing IC temperatures has the negative effect of increasing short-circuit power indirectly through $I_{\text{short-circuit}}$.

The transistor leakage power has three major components: reverse-biased junction leakage current, gate direct tunneling leakage and sub-threshold leakage with the major component being the sub-threshold leakage [69]. The sub-threshold current is

$$I_{\text{sub}} = I_{\text{s0}} \exp \frac{V_{gs} - V_{th}}{nV_T} \left(1 - \exp \frac{-V_{ds}}{V_T}\right) \tag{3.4}$$

where V_T is the thermal voltage, I_{s0} is a constant that is device dependent and n is a constant [61]. The thermal voltage is calculated as follows:

$$V_T = kT/q \tag{3.5}$$

where k is the Boltzmann constant, q is the charge of an electron and T is the absolute temperature in Kelvin.

The amount of heat generated from each source ($P_{\text{dynamic}}, P_{\text{short-circuit}}, P_{\text{leakage}}$) varies from circuit to circuit and depends on factors such as the technology used, the clock frequency, vdd voltage etc.

3.1.2 Interconnect Joule Heating

Heat is also generated in ICs by transporting current through the interconnect to the transistors. This heat is created due to the interconnect resistance and is referred to as Joule heating. The Joule heating power at a given time, $J(t)$, in each wire/interconnect segment is

$$J(t) = i^2(t) R \tag{3.6}$$

where R is the resistance of the interconnect segment and $i(t)$ is the current through the segment.

In a typical IC most of the Joule heating occurs in the Power Supply Network (PSN) since all the energy required for the circuit flows through it. Within a PSN Joule heating occurs in the wires, power supply bumps, and vias between layers in greatly differing magnitudes. Figure 3.1 shows the transient Joule heating power of the wires and pads for the ibmpg1 PSN benchmark [35]. The Joule heating in the power supply pads stays

roughly constant as soon as the PSN approaches its steady state response. However, the Joule heating of the wires has spikes corresponding to the transistors switching which leads to a much larger Joule heating RMS value of 2.3W in the wires compared to 0.6W in the power supply pads.

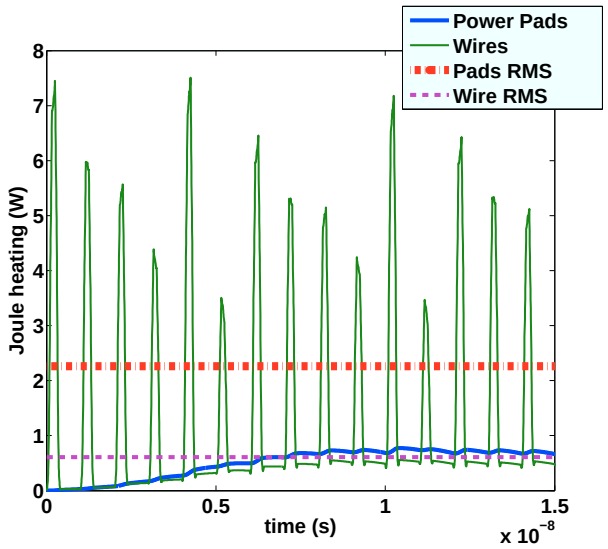


Figure 3.1: An example from the ibmpg1 transient benchmark shows that Joule heating in the wires has a much larger RMS value than the power supply pads.

The amount of Joule heating varies across the different metal layers in a typical PSN, with the lowest metal layer having the largest Joule heating RMS power due to the large current spikes on this metal layer as a result of its' proximity to the switching transistors. Figure 3.2 shows the distribution of Joule heating across the various metal layers in the ibmpg2 transient benchmark. Most Joule heating occurs on the lowest metal layer (1.61W RMS), which is significant since this metal layer has the smallest wire dimensions and is consequently more susceptible to electromigration. The smaller width of the lower metal wires leads to less heat diffusion to the substrate and consequently higher wire temperatures. In addition, the proximity to the substrate means that heat from the device switching

activity likely already increases the temperature (and therefore decreases MTTF) of the wire due to T_{sub} in Equation 3.13.

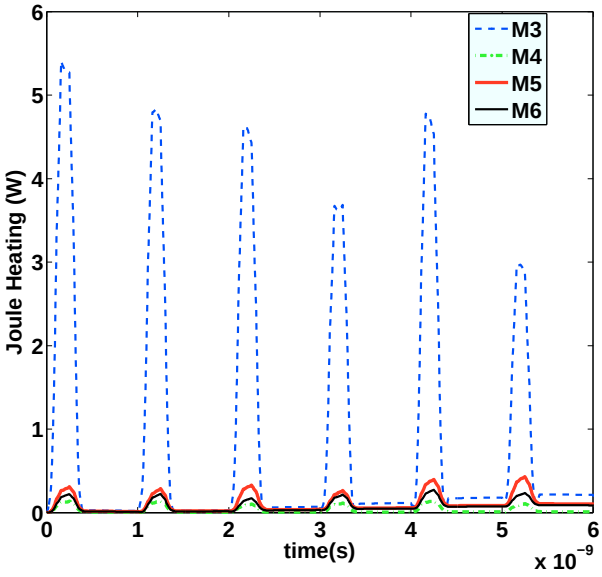


Figure 3.2: An example from the ibmpg2 transient benchmark shows that lower metal layers suffer from larger Joule heating than global metal layers.

The RMS Joule heating values for the M5 and M6 wires are 0.18W and 0.14W, respectively, which is not as large as the value for the M3 layer. However, the impact of Joule heating is still significant since the M5 and M6 layers are located farther from the heat sink and consequently have much larger thermal resistance to the substrate which leads to higher temperatures. The Joule heating in the vias between metal layers is insignificant in comparison to the Joule heating in the wires and pads (RMS value of only 0.0009W) as illustrated in Figure 3.3.

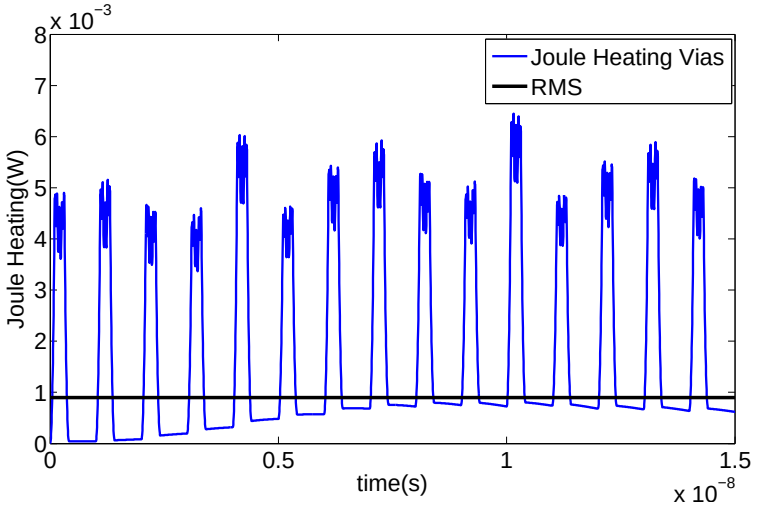


Figure 3.3: An example from the ibmpg2 transient benchmark shows that Joule heating in the vias between different metal layers is not very significant (1×10^{-3}W).

3.2 Substrate Temperature Modeling

The temperature of an IC is governed by the rate of heat generation and removal. Most of the heat generated in ICs flows from the transistor junction through the substrate, to the heat spreader, to the package, and is finally removed through the heat sink to the ambient air. Some of the heat, although not the majority, flows in the opposite direction toward the package and the package bumps which are in contact with the PCB and the ambient air. The rate at which heat is removed is governed by the heat conduction equation

$$\rho c_p \frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot [k(\vec{r},T)\nabla T(\vec{r},t)] + g(\vec{r},t) \tag{3.7}$$

where g is volume power density (W/m^3), T is temperature (K), c_p is specific heat capacity (J/KgK), ρ is the density of the material and k is the thermal conductivity of the material.

The boundary conditions for Equation 3.7 are usually assumed to be as follows:

The vertical sides of the chip, and the side not attached to the heat sink are assumed to be adiabatic. The side of the chip attached to the heat sink is assumed to be convective and is modeled using the following equation [69]:

$$k\left(\vec{r},T\right)\frac{\partial T\left(\vec{r},t\right)}{\partial n_i}=h_i\left(T_a-T\left(\vec{r},t\right)\right) \tag{3.8}$$

For homogeneous materials (whose thermal conductivity is temperature independent) at steady state, Equation 3.7 can be simplified to

$$-k\nabla^2 T=g \tag{3.9}$$

Due to size and complexity of most modern ICs, Equation 3.9 is usually applied at the block level for thermal modeling. Consequently g corresponds to the average power density of the transistors that comprise the block, T is the temperature of the block, and k is the combined thermal conductivity of the materials which separate the transistors from the heat sink, usually bulk silicon, but this may vary depending on the specific manufacturing for the IC (for example SOI).

The temperature map for an IC is obtained by solving Equation 3.7 using several methods such as Finite Element Method [74], Finite Difference [89, 98], Random walk [100] or Green-based methods [30,107,108]. Figure 3.4 shows a thermal map for the GSRC n100 floorplanning benchmark circuit and highlights the non-uniformity of the temperature map on modern ICs with large hotspots and temperature gradients.

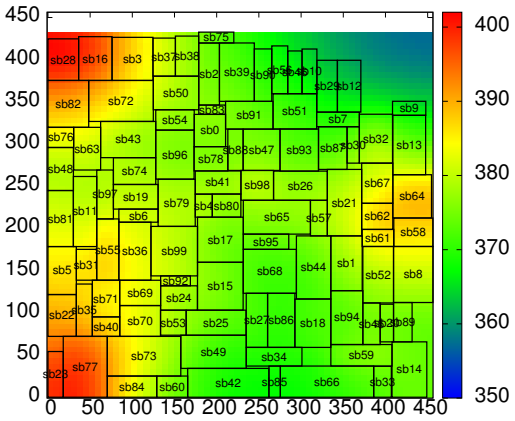


Figure 3.4: IC temperature map for GSRC n100 circuit showing temperature hotspots. The circuit dimensions are in μm and the temperature is in Kelvin (K)

3.2.1 Thermal Modeling of Interconnect

IC interconnect temperatures are governed by the substrate temperature of the IC in the vicinity of the interconnect and the quantity of Joule heating in the interconnect. Joule heating leads to the interconnect experiencing a temperature increase, ΔT_{wire}, which is proportional to the root mean square (RMS) value of the Joule heating power [6] and is computed using

$$\Delta T_{\text{wire}} = \frac{RR_\theta}{D} \int_0^D i^2(t)\,\mathrm{d}t. = i^2_{\text{RMS}}\,(t)\,RR_\theta \tag{3.10}$$

where i_{RMS} is the RMS value of the current, R_θ is the thermal resistance of the interconnect to the substrate, R is the resistance of the interconnect, and D is the duration over which the Joule heating is being analyzed.

The thermal resistance, R_θ, is measured with respect to the substrate since the majority of heat produced in an IC is removed from the heat sink attached to the back-side

of the substrate as stated in Section 3.2. This thermal resistance can be estimated [6] as

$$R_\theta = \frac{t_{\text{ins}}}{K_{\text{eff}} L W_{\text{eff}}} \tag{3.11}$$

where t_{ins} is the thickness of the insulation between the metal interconnect and the substrate, K_{eff} is the effective thermal conductivity of the thermal insulation, L is the length of the interconnect and W_{eff} is the effective width of the interconnect. The interconnects farthest from the substrate are more susceptible to Joule heating due to their large t_{ins}. In addition, interconnects located close to the substrate tend to be smaller in dimension and consequently have a larger R_θ due to their smaller width and area to diffuse heat. The K_{eff} term takes into consideration the thermal conductivity of various metal, insulator and via materials between the substrate and the interconnect. Since Equation 3.11 is based on a 1-D thermal diffusion model, W_{eff} is used to model heat lost in other dimensions [6] and is calculated as

$$W_{\text{eff}} = W_m + \phi t_{\text{ins}}. \tag{3.12}$$

where W_m is the width of the interconnect and ϕ is a heat spreading factor.

The final temperature of each interconnect element depends on both the global temperature due to dynamic switching and the corresponding heat from the substrate along with the local Joule heating according to

$$T_{\text{wire}} = T_{\text{sub}} + \Delta T_{\text{wire}} \tag{3.13}$$

where T_{sub} is the temperature due to the substrate directly below the interconnect and ΔT_{wire} was defined in Equation 3.10.

The effect of Joule heating on interconnect temperatures is illustrated in Figure 3.5(a) which contains a histogram showing the interconnect ΔT caused by Joule heating for the ibmpg1 benchmark.

The temperature increase from Joule heating is usually small, however, it is highly skewed with numerous interconnects having $\Delta T > 10K$ and a few interconnects having $\Delta T > 30K$. The actual distribution of interconnect temperatures is less skewed than that of ΔT (Figure 3.5(b)) since the substrate temperature evens out the distribution slightly.

3.3 Thermal Reliability Issues

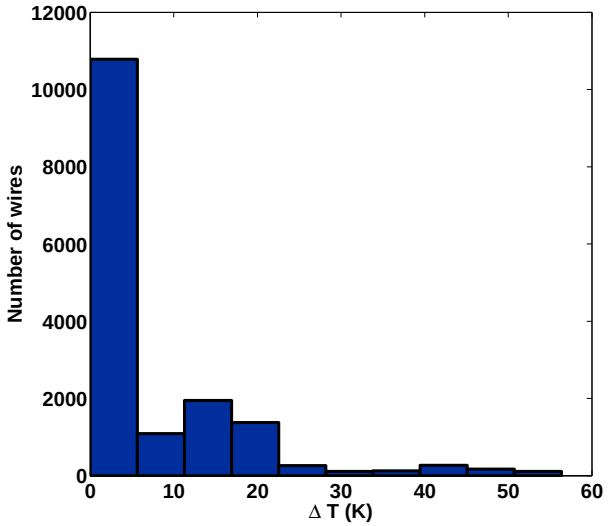
Large temperatures can significantly affect the reliability of an IC in several ways, such as increased leakage currents, increased threshold voltages due to NBTI, increased electromigration, increased resistivity and increased bump failure to thermal cyclic fatigue.

3.3.1 Negative Bias Temperature Instability

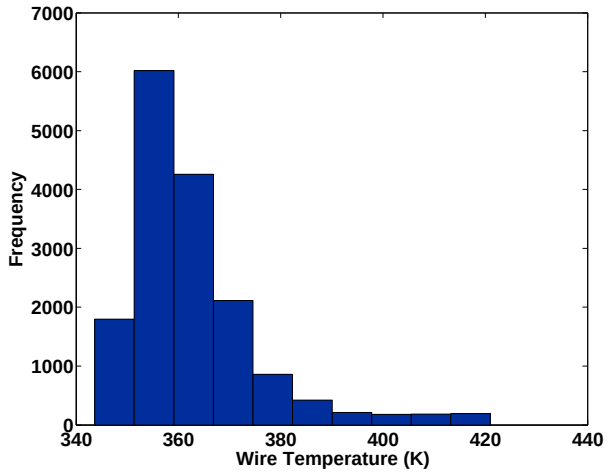
Negative Bias Temperature Instability (NBTI) is the degradation of the threshold voltages, drive currents and noise margins in negative bias transistors (usually PMOS) as a result of interface traps created by broken Si-H bonds [4,20,93]. The difference in threshold voltage from NBTI is calculated as follows [93]:

$$\Delta V_{th} = \frac{qN_{it}}{C_{ox}} \text{ , where } C_{ox} = \frac{\epsilon_{ox}}{T_{ox}} \tag{3.14}$$

where N_{it} represents the number of interface traps and is estimated using the following model [93]



(a) Most interconnects experience small increases in temperature, but a few critical interconnects experience large increases in temperature.



(b) Most interconnects temperatures are around the same value

Figure 3.5: Distribution of ΔTemperatures and actual Temperatures of wires for the IBMpg1 benchmark showing the skewed ΔTemperatures and less skewed wire temperatures.

$$\Delta N_{it} = \left(K^2 \cdot t^{0.5} + c^{\frac{1}{2n}}\right)^{2n} \tag{3.15}$$

with K being exponentially proportional to temperature as follows [93]:

$$K \propto \sqrt{C_{ox}\left(V_{gs} - V_{th}\right)} \cdot \exp(E_{ox}/E_o) \cdot \exp(-E_o/kT) \tag{3.16}$$

Equation 3.16 shows the strong exponential relationship between N_{it} and temperature. Increasing temperatures increases the number of interface traps forming which results in massive decreases in voltage threshold (V_{th}). Increased threshold voltages lowers drain current which has the potential to lead to timing errors.

3.3.2 Leakage Currents

There are three main types of leakage currents in CMOS circuits: reverse-biased junction leakage current, gate direct tunneling leakage and sub-threshold leakage. All of these various leakage currents have strong dependence on temperature, with sub-threshold leakage have an exponential relationship.

The sub-threshold leakage current (I_{sub}) for a transistor is the drain-source current when the transistor is operating in the weak inversion region. The I_{sub}, defined in Equation 3.4, can also be represented using the follow equation:

$$I_{\text{sub}} = k_{\text{tech}} \left(\frac{W}{L}\right) 10^{-\frac{V_{th}q}{2.3nkT}} \tag{3.17}$$

where k_{tech} is a transistor geometry and CMOS technology dependent parameter, W and L represents the transistor width and length respectively, V_{th} represents the threshold voltage, q represents the charge of an electron, k is the Boltzmann constant, T is the absolute

temperature in Kelvin and n is a constant greater than 1 that is device dependent [69]. Equation 3.17 illustrates the exponential dependence of the sub-threshold leakage current on temperature. The magnitude of $I_{\rm sub}$ increases on average between 8-12x/100° C [69]. Thus, large on chip temperatures significantly increase leakage power which can account for more than 50% of total power at modern technology nodes [61].

The gate direct tunneling leakage current flows from the gate to the substrate through the insulator and is modeled as

$$I_{\rm gate} = \alpha \exp(-\beta T_{ox}) W \tag{3.18}$$

where W is the width of the transistor, T_{ox} is the oxide thickness and α and β are temperature dependent constants specific to a technology [24]. Gate leakage has steadily become a concern in modern ICs due to the decreasing levels of oxide thickness. The sensitivity of $I_{\rm gate}$ to large on chip temperatures is not as significant as $I_{\rm sub}$ and $I_{\rm gate}$ only increases about 2x/100° C [69].

The junction leakage current flows from the source or drain to the substrate when the transistor is off and is modeled as

$$I_{\rm rev} = I_s \left(\exp \frac{V_{dd}}{v_{th}} - 1 \right) \tag{3.19}$$

where V_{dd} is the transistor voltage, and I_s is a device dependent constant that depends on the area and perimeter of the diffusion region, and the doping concentration [29]. I_s is strongly temperature dependent and the junction leakage can increase by factors of 50-100x/100° C. However for low temperatures (below 150 ° C), junction leakage is negligible [69].

3.3.3 Thermal Effects on Interconnect Resistivity

IC interconnect resistance is directly proportional to temperature, hence, increases in wire/bump temperature negatively affect the reliability of the circuit as a result of the increase in bump/wire resistance and potential IR drop/voltage droop problems.

The resistance of a wire/bump is calculated as

$$R = \rho \frac{l}{A} \tag{3.20}$$

where ρ is the resistivity of the wire/bump, l is the length of the wire/bump and A is the cross-sectional area of the wire/bump perpendicular to the current flow. The resistivity of a wire/bump is dependent on temperature and is calculated as

$$\rho\left(t\right) = \rho_o\left(1 + \alpha_r\left(T - T_o\right)\right) \tag{3.21}$$

where T_o is a reference temperature, ρ_o is the resistivity at T_o, T is the wire/bump temperature and α_r is the temperature coefficient of resistivity.

Combining Equation 3.21 and Equation 3.20, the change in interconnect resistance (ΔR) due to a temperature increase is

$$\Delta R = \frac{1 + \alpha\left(T_{new} - T_o\right)}{1 + \alpha_r\left(T_{old} - T_o\right)} \tag{3.22}$$

where T_{old} to T_{new} are the initial and final wire/bump temperature respectively and T_o is the reference temperature.

The impact of the temperature increase on interconnect reliability can be quite significant as illustrated in Figure 3.6. The resistance of some wires in the ibmpg1t benchmark are increased to $1.18\times$ of their original value. The large effect of wire temperatures on interconnect resistance highlights the need to control Joule heating in wires.

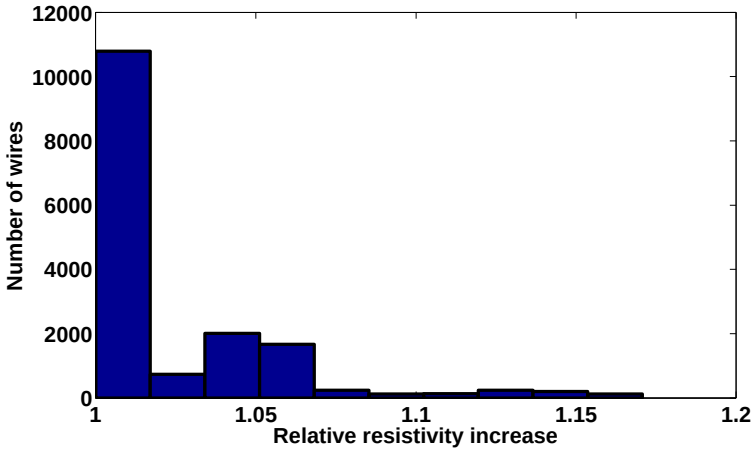


Figure 3.6: Most wires experience a small change in resistance, but a few critical wires experience large increases in resistance, up to 1.18 times the original value.

3.3.4 Thermal Effects on Electromigration

Electromigration has steadily become a concern in the design of power supply networks due to the significant increases in the power demanded by modern ICs [57]. The Mean Time to Failure (MTTF) of a metal wire/bump due to electromigration is calculated using Black's Equation

$$\text{MTTF} = A\frac{1}{j_{\text{avg}}^n}\exp\left(\frac{Q}{kT}\right), \tag{3.23}$$

where A is a constant based on the cross-sectional area of the wire/bump, j_{avg} is the average current density ($\frac{A}{cm^2}$), Q is the activation energy, n is a fitted model parameter, k is the Boltzmann's constant, and T is the wire/bump temperature.

The electromigration failure rate is exponentially dependent on temperature hence even small increases in temperature produce large decreases in electromigration reliability. If the temperature of wire/bump changes from temperature T_{old} to T_{new}, the MTTF failure

rate decrease is calculated from Equation 3.23 as

$$\Delta\text{MTTF} = \exp\left(\frac{Q}{k}\left(\frac{1}{T_{old}} - \frac{1}{T_{new}}\right)\right). \tag{3.24}$$

This equation assumes that the j_{avg} through the wire/bump remains constant.

The impact of the temperature increase on interconnect reliability can be quite significant as illustrated in Figure 3.7. The electromigration lifetime of some wires are decreased to $0.2\times$ of their original values in the ibmpg1t benchmark. The large effect of wire temperatures on interconnect electromigration lifetime highlights the need to control Joule heating in wires.

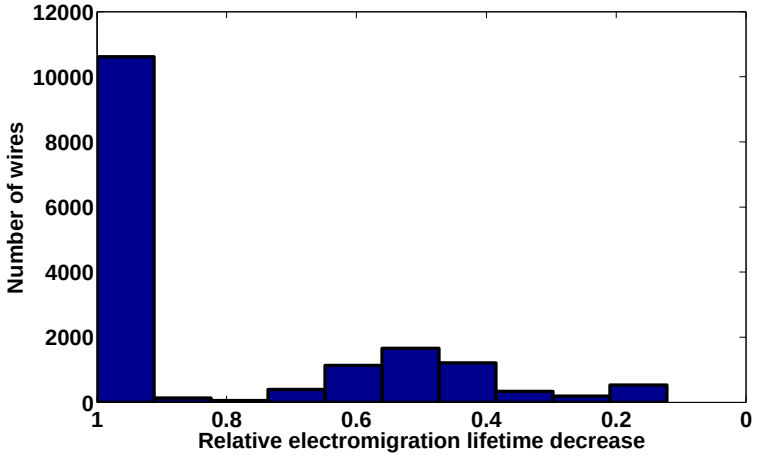
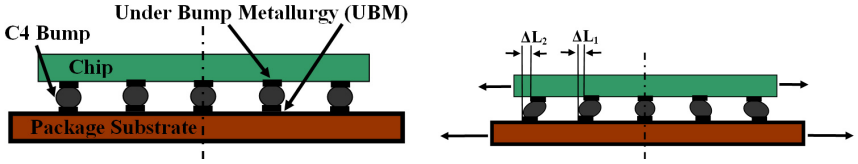


Figure 3.7: Most wires experience a small decrease in electromigration lifetime. However a significant amount of wires have their lifetime reduced to 0.5 of the original value and some have as high reductions as 0.2 of the original value.

3.3.5 Thermal Effects on Package Reliability

Thermal-Mechanical stresses are an unwanted byproduct of the large temperatures created in modern ICs and are caused by the Coefficient of Thermal Expansion (CTE) mismatch between the silicon chip and the substrate. The CTE mismatch causes the chip

and the substrate to expand by different amounts when subjected to changes in temperature which results in shear strains in the solder bumps as shown in Figures 3.8(a)and 3.8(b). Figure 3.8(a) shows the substrate and chip in the equilibrium state and Figure 3.8(b) shows the substrate and chip after an increase in temperature. These strains produce stresses which leads to mechanical fatigue over a certain amount of heating and cooling cycles which correspond to the on and off state of the IC. A recent $150-200MM recall by NVIDIA [63] of certain GPUs due to solder bump failure from thermal-cycling stress fatigue illustrates the severity of the problem.



(a) Equilibrium Bump Positions with no strain

(b) Bump Positions after Temperature Increase showing large strains especially in edge bumps

Figure 3.8: CTE mismatch causes large strains in C4 bumps.

The problem is only going to worsen as the industry moves to lead-free C4 solder bumps such as those made from SnAgCu due to recent legislature banning the usage of hazardous lead in electronic components. Solder bumps made from this material and other lead-free solders are susceptible to low cycle thermal fatigue failure as a result of their viscoplacsiticity. Failure in C4 bumps is usually due to fracture which is caused by crack formation and propagation caused by cyclic thermal induced stresses. Figure 3.9 illustrates

the magnitude of the cracks that can be created in lead-free C4 solder bumps from thermal-mechanical stresses [22].

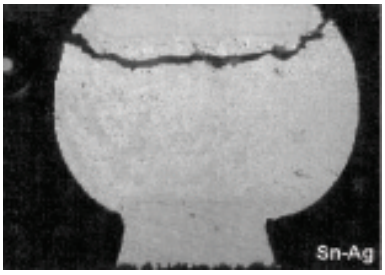


Figure 3.9: Complete fracture of a SnAg Solder ball from thermal-cyclic fatigue. [22]

3.3.5.1 Stress and Strain

The magnitudes of the thermally induced stresses depend on the amount of shear strain experienced in the C4 bumps. The shear strain caused by the CTE mismatch is approximated using the following formula [22]

$$\gamma = r\left(\alpha_s - \alpha_c\right)\Delta T \tag{3.25}$$

where r, is the scaled distance from the centroid of chip/substrate, α_s is the CTE for the substrate, α_c is the CTE for the chip, ΔT is the difference in temperature from the equilibrium temperature value and γ is the shear strain. It should be noted that the maximum shear strain occurs at the edge of the chip and no shear strain occurs at the center as illustrated in Figure 3.8(b).

Equation 3.25 assumes that there is no underfill between the substrate and the chip. Underfill provides mechanical reinforcement between the die and substrate, consequently reducing the strains and stresses experienced by C4 balls. Modeling the underfill

effects, however, is a complex task and usually requires Finite Element Analysis (FEA). These effects are ignored in order to accelerate the time taken to calculate the strain and stresses in C4 balls. FEA simulations require significant computing time and consequently greatly increases algorithmic runtimes, especially if multiple simulations are required such as in an iterative algorithm. Since the underfill effects are ignored, the strains estimated by the proposed model are an upper limit to the actual value.

The thermal cyclic stresses in the C4 bumps are calculated from the equivalent strain experienced by the bumps. Assuming that the C4 solder material obeys Von Mises' yield criterion, the equivalent strain is approximated as

$$\begin{aligned} \epsilon_e &= \frac{\sqrt{2}}{3}[(\epsilon_{xx}-\epsilon_{yy})^2+(\epsilon_{yy}-\epsilon_{zz})^2+(\epsilon_{zz}-\epsilon_{xx})^2 \\ &\quad +\frac{3}{2}(\gamma_{xy}^2+\gamma_{yz}^2+\gamma_{xz}^2)]^{\frac{1}{2}} \end{aligned} \tag{3.26}$$

where ϵ and γ are the various normal and shear strains in the solder material [90]. Von Mises' yield criterion is used to determine the shear stresses at which metals begin to yield and occurs when the second deviatoric stress invariant reaches a critical value [66].

Since the dominant shear strain in solder balls is in the xy plane, the normal and shear strains are set to 0 except γ_{xy}. The equivalent stress is then estimated using the stress-strain relationship

$$\sigma_e = E\epsilon_e \tag{3.27}$$

where σ is the equivalent stress and E is the Young's modulus of the material.

The Young's modulus of solder is, however, temperature dependent, hence it is approximated as [90]

$$E = 52708 - 67.14T - 0.0587T^2. \tag{3.28}$$

3.3.5.2 Creep

The rate of failure from thermal-cyclic fatigue is primarily dependent on the amount of creep in the C4 balls. Creep is defined as the slow deformation of a material subject to high stresses. It is positively correlated to temperature in that large temperatures increase the rate of the creep deformation. The creep strain rate in SnAgCu C4 solder balls is calculated by using the Garofalo-Arrhenius hyperbolic sine law

$$\frac{d\gamma}{dt} = C\left(\frac{G}{\Theta}\right)\left[\sinh\left(\omega\frac{\tau}{G}\right)\right]^{n}\exp\left(-\frac{Q}{k\Theta}\right) \tag{3.29}$$

where $\frac{d\gamma}{dt}$ is the creep shear strain rate, γ is the creep shear strain, C is a material constant, Θ is the absolute temperature, G is the shear Modulus, τ is the shear strain, Q is the activation energy, k is Boltzmann's constant, n is the stress exponent and ω is the stress level [44]. Since the C4 bumps obey Von Mises' criterion, Equation 3.29 can be rearranged to:

$$\frac{d\epsilon}{dt} = C_1\left[\sinh\left(C_2\sigma\right)\right]^{C_3}\exp\left(-\frac{C_4}{T}\right) \tag{3.30}$$

where $C1$, $C2$, $C3$, and $C4$ are material constants, σ is the equivalent stress, T is temperature, $\frac{d\epsilon}{dt}$ is the equivalent creep strain rate [44]. The equivalent stress is calculated using Equation 3.27. The temperature of the bumps are determined by two factors: 1) The self-heating generated in the bumps as current flows through them and 2) The temperature of the blocks within the vicinity of the bumps, thus bumps that carry large currents (power supply network bumps) and bumps located in hotspots of the IC are more susceptible to failure from thermal-cyclic fatigue.

The creep strain rate can be converted to a creep strain range which is inversely

proportional to number to cycles to thermal cyclic failure. The conversion is as follows:

$$\Delta\epsilon_c = t\dot{\epsilon_c} \tag{3.31}$$

where $\dot{\epsilon_c}$ is the creep strain rate and t is the length of time a bump is subjected to a high stress (i.e., the time the chip is in an active state causing high temperatures and consequently high stresses).

3.3.5.3 C4 Ball Failure

There are several methods of estimating the mean cycles to failure for C4 solder balls, however the Knecht-Fox model is used throughout this thesis due to its simplicity and accuracy [65]. The model is described in the following equation:

$$N_f = \frac{C}{\Delta\epsilon_c} \tag{3.32}$$

where N_f is the number of cycles to failure, C is an empirical constant and $\Delta\epsilon_c$ is the creep strain range [65]. The creep strain range is directly proportional to the creep strain rate (Equation 3.31) which is exponential dependent on temperature (Equation 3.30) hence increasing IC temperatures will lead to more C4 bumps failing from thermal cyclic fatigue.

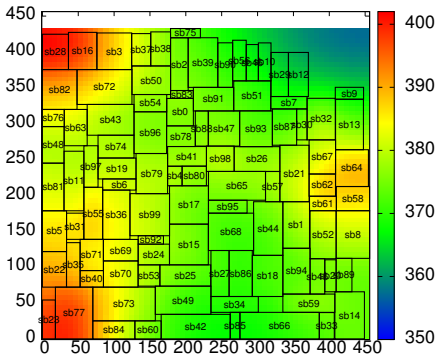
Chapter 4

Improving IC Temperatures through Floorplanning

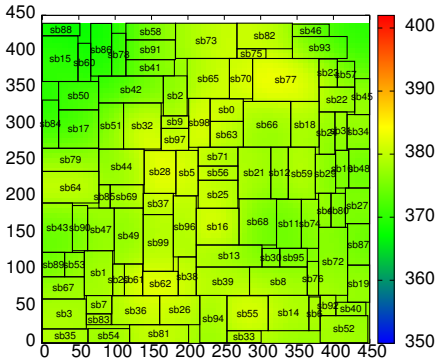
The high temperatures found in modern ICs can be reduced by thermal-aware floorplanning. Figure 4.1 shows two floorplans for the same circuit, however the maximum temperature in one circuit is 19K larger than the other, with the only difference between the circuits is the placement of the circuit blocks and the routing between blocks. This chapter of the thesis discusses the factors that affect chip cooling that can be controlled during floorplanning, then introduces methods of incorporating them during floorplanning.

4.1 Floorplanning Influence on Overall Chip Temperature

There are two ways in which floorplanning can be used to control the final temperature of a circuit: 1) adjusting block power densities and 2) reducing thermal coupling between high temperature blocks by moving them apart. This section highlights the relationships between chip temperatures and these two factors, and introduces ways to control



(a) n100 Floorplan 1 with Maximum Temperature of 401K. The Circuit dimensions are in μm and the temperature is in Kelvin (K)



(b) n100 Floorplan 2 with Maximum Temperature of 382K. The Circuit dimensions are in μm and the temperature is in Kelvin (K)

Figure 4.1: Example floorplans from n100 showing how block layout can affect peak temperature.

these factors during floorplanning.

4.1.1 Temperature and Power Density

The temperature of a block is strongly correlated to its power density as can be seen by analyzing Equation 3.9. This phenomena can be demonstrated with a simple experiment where an isolated block is kept at a constant area, but has its power density slowly increased. The results of such a thermal simulation experiment (conducted using Hotspot [82]) is shown in Figure 4.2. The relationship between power density and temperature is linear for an isolated block not exposed to any thermal coupling or boundary conditions.

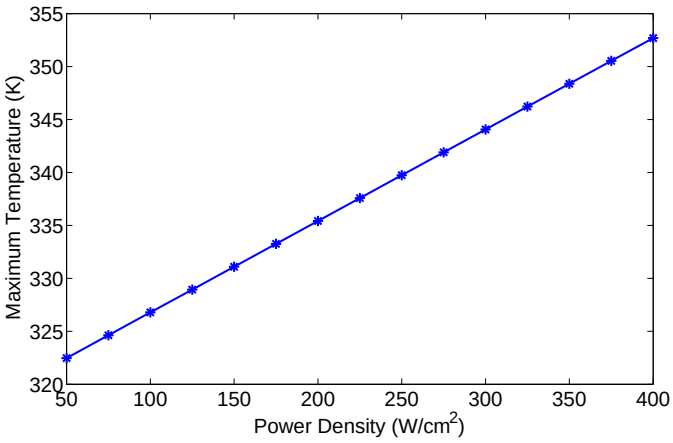
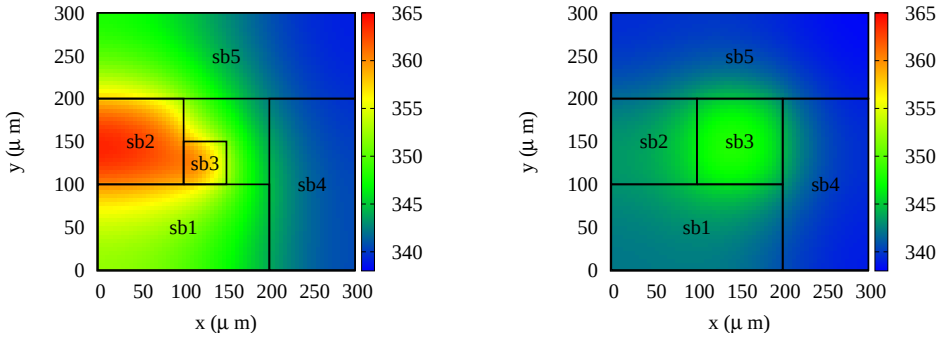


Figure 4.2: Maximum temperature vs power density for an isolated block showing a linear relationship.

The power density of circuit blocks can be adjusted during floorplanning by changing their area utilization. They are two types of blocks used in floorplanning, soft blocks and hard blocks. Soft blocks are typically generated from synthesized logic and consequently have variable aspect ratios and areas. The area of soft blocks is typically de-

cided by routing congestion; blocks need to have enough whitespace to be routed. However, since power density is inversely proportional to area, soft blocks that have a high switching activity should be given slightly more area, than what is required for routing to reduce maximum chip temperatures. Figure 4.3 shows an example where the temperature of a chip is altered by adjusting the utilization of a soft hot block.

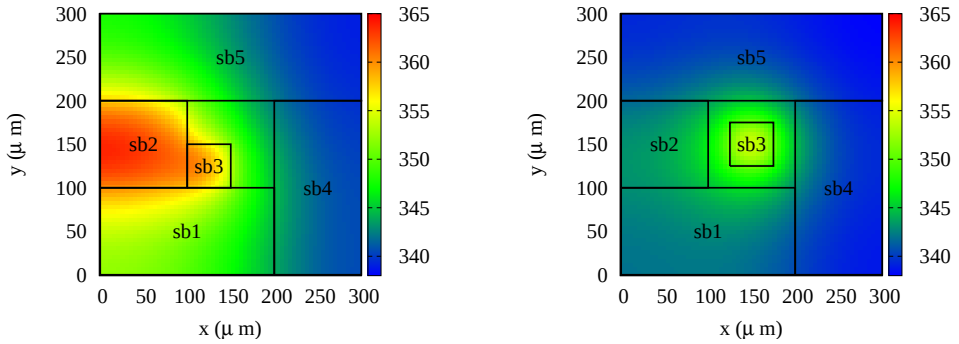


(a) Floorplan without soft block area utilization: Max Temp is 363.8K

(b) Floorplan with soft block area utilization: Max Temp is 348.1K

Figure 4.3: Soft block area utilization can reduce maximum temperatures significantly.

Hard blocks are typically IP blocks and consequently have a fixed area and fixed aspect ratio. However, the virtual power density of these blocks can be changed during floorplanning by adding whitespace around the blocks (creating a virtual block) which would also reduce the thermal coupling with other blocks. Figure 4.4 shows an example where the temperature of a chip is altered by adjusting the whitespace around a hard hot block. It should be noted that enlarging soft blocks is more effective at reducing temperatures than enlarging the whitespace around hard blocks.



(a) Floorplan without hard block area utilization: Max Temp is 363.8K

(b) Floorplan with hard block area utilization: Max Temp is 354.1K

Figure 4.4: Hard block area utilization can reduce maximum temperatures also.

4.1.2 Temperature and Thermal Coupling

The temperature of a block is also influenced by its surrounding blocks. If two hot blocks are in close proximity, the maximum temperature for the chip might be higher than if blocks were spread apart. This phenomena occurs because of thermal coupling between the two hot blocks. When blocks dissipate power the primary direction of heat flow is in vertical direction, however heat flows in the other cardinal directions also. The amount of heat flow is determined by the difference in temperature between the block under consideration and the neighboring blocks. If the difference in temperature is small, there is less heat flow in the other cardinal directions which results in higher temperatures.

Similarly, since the edge of the chips are almost adiabatic, hot blocks placed close to the edge of a chip will have higher temperatures than if they were placed closer to the center. It should also be noted that blocks with the same power density and different areas

will have different temperatures due to the fact that a larger block will have less thermal diffusion in its center than a smaller block. As a result the larger block will have a higher temperature than the smaller block.

These phenomenon are illustrated by the use of simple IC thermal simulation experiments. The effects of thermal coupling between blocks on chip temperatures is illustrated in Figure 4.5 which shows the result of an experiment where the distance between two high power-density blocks is slowly incremented. The results show that spreading high power density blocks can reduce high on chip temperatures.

The effects of the adiabatic nature of the chip edges on IC temperatures is illustrated in Figure 4.5. In this experiment the distance of a high power density block from the edge of the chip is incrementally increased. The results show that block temperatures can also be decreased by spreading high power density blocks from the edge of the chip which is congruent with the results of [45].

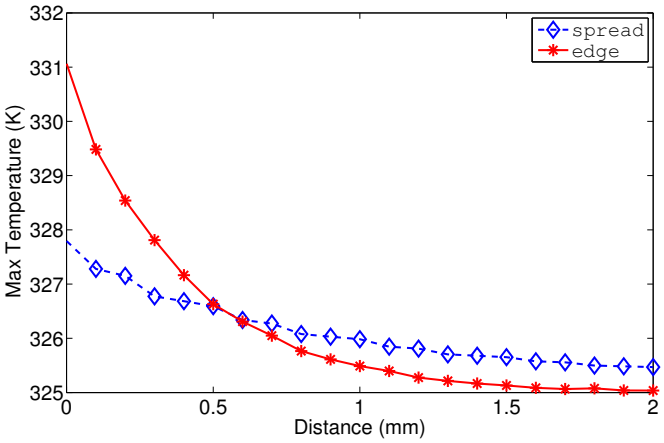


Figure 4.5: Maximum temperature vs distance from edge for a 100 W/cm^2 block (red) showing decrease in maximum temperature as block's distance from the edge increases. Maximum temperature vs separation distance for two 100 W/cm^2 (blue) blocks showing decrease in maximum temperature as the blocks are moved further apart.

The effects of block sizes on chip temperatures is illustrated in Figure 4.6. In this experiment the area of a 200 W/cm^2 block is incrementally increased. The results shows block temperatures are indeed dependent on area which is congruent with the results found in [72].

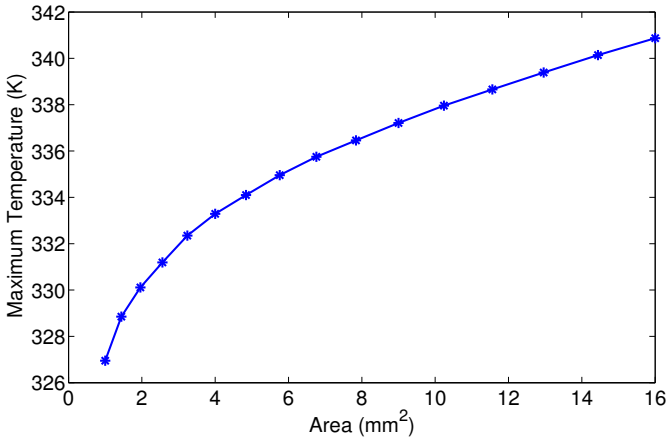


Figure 4.6: Maximum temperature vs area for a 200 W/cm^2 block showing that the maximum temperature is also dependent on area and not only power density.

4.2 Proposed Thermal Floorplanning Techniques

This section introduces the novel floorplanning moves that reduce block power densities and thermal coupling between hot blocks. The floorplanning moves assume a simulated annealing methodology, but some of the moves are applicable to other floorplanning methodologies, such as force-directed approaches [26].

4.2.1 Powerspreading Cost to Reduce Thermal Coupling

Reducing thermal coupling between high temperature blocks is an effective way of reducing high temperatures in circuits as shown in Section 4.1.2. One method to prevent thermal coupling is to create a cost function that is minimal when high temperature blocks are optimally separated from each other and the adiabatic boundary countries of the chip. Inspired by statistical mechanics, such a cost function is created by using two sets of short range repulsion equations that govern the interactions of high power density blocks to evenly space them apart and to space them away from the edge of the chip.

4.2.1.1 Block Repulsion Force

The first set of short range repulsion equations is the block repulsive force (P_B) which is used to maximally spread high power density blocks apart. The equation to calculate the average block repulsive force is

$$P_B = \frac{1}{n} \sum_{i,j}^{n} \frac{p_i + p_j}{d_{ij}^2} \tag{4.1}$$

where p_i is the power of block i, d_{ij} is the Euclidean distance between the edges of block i and block j, and n represents the number of blocks in the floorplan that have a power density greater than a specified value.

Not all blocks are considered for two reasons. Firstly, limiting the number of blocks reduces the runtime required for calculating the block repulsive force. Secondly, the hottest blocks in a floorplan, usually have a comparatively large power density value with respect to the other blocks in the floorplan. Consequently, it is sufficient to just consider those blocks for power spreading since a block that has a relatively smaller power density

will never become the hottest block.

It should also be noted that even though power density is used to select which blocks to consider for power spreading, the cost function uses the power of the blocks to calculate the cost. By doing this, the area of a block is taken into consideration. Otherwise, two small high power density blocks spaced a fixed distance apart would have the same cost as two large blocks with the same power density spaced at the same distance, even though there is significantly more thermal coupling between the two large high power density blocks.

4.2.1.2 Edge Repulsion Force

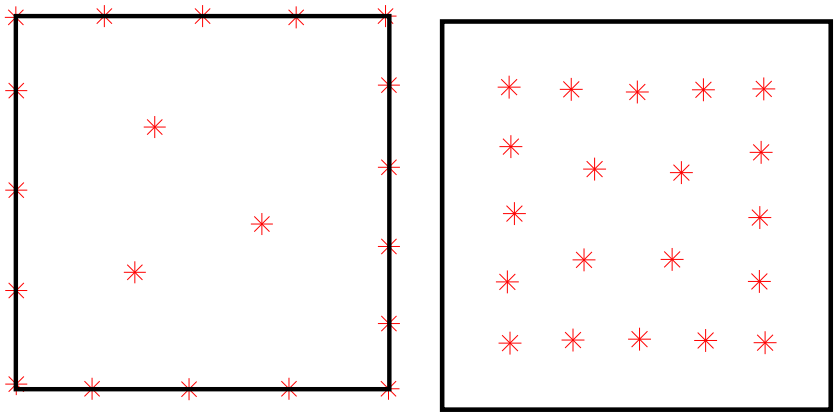
The second set of short range repulsion equations are the edge repulsive forces (P_E) which are used to push the hot blocks away from the edge of the chip. The edge repulsion force is calculates as

$$P_E = \sum_{i}^{n} \left(\frac{p_i}{x_i^2} + \frac{p_i}{y_i^2} \right) \tag{4.2}$$

where x_i is the smallest distance in the x direction of block i from the edge of the chip and y_i is the smallest distance in the y direction of block i from the edge of the chip.

If only the P_B forces are used, many of the blocks will be placed at the edge of the chip which can increase peak temperature as shown in Section 4.1.2. This is illustrated by a simple experiment using 20 blocks where the cost function only considers the P_B forces. The results of the experiment, displayed in Figure 4.7 (blocks shown as asterisks) shows many blocks being forced to the chip's edge if P_E is not considered (Figure 4.7(a)). If the cost function also considers P_E, blocks are pushed from the edge as shown in Figure 4.7(b)

resulting in lower chip temperatures.



(a) Edge Cost = 0 showing blocks congregating at the edge of the chip

(b) Edge Cost = 10 showing blocks spread apart and also away from the edge of the chip

Figure 4.7: Simple floorplan showing the necessity of including an edge cost to move blocks from the edge of a chip

The final power spreading cost is

$$S_P = P_B + c \times P_E \tag{4.3}$$

where c determines the contribution of edge and weight repulsion forces. Changing c adjusts how close hot blocks will get to the edge of the chip. Large values results in too many blocks congregated in the center and small values of c, result in blocks being too close to the edge.

4.2.1.3 Advantages of Using a Power Spreading Cost

The main advantage of using a power spreading cost is the reduced runtime required for thermal floorplanning. The main bottleneck of thermal floorplanning is calculating block temperatures since solving Equation 3.7 accurately requires significant computation time. Most previous researchers have developed alternate methods of computing block temperatures for chips [67, 94, 96] which are still slow compared to other metrics evaluated during floorplanning. Some researchers [15, 34, 102] have abandoned temperature simulations altogether and have used an approximated linear 1-D form of Equation 3.7 to approximate block temperatures using power densities. Such methods tend to lead to very inaccurate temperature simulations and consequently suboptimal floorplans. Evaluating the proposed power spreading cost, is quick with respect to evaluating the other floorplanning metrics such as area and HPWL, but does not sacrifice solution quality since the cost function is minimal when hot blocks are separated from each other and the edge of the chip.

4.2.2 Whitespace Allocation to Lower Block Power Densities

One of the major drawbacks of previous thermal-aware is that there is only one method of reducing temperatures, separating high temperature blocks. Area utilization is an effective method or reducing block power densities and consequently block temperatures in circuits as shown in Section 4.1.1. However, there is a limit on the amount of area utilization that can be done since there is a limited amount of whitespace available in the floorplan. Two methods are proposed for allocating the available whitespace: 1) statically allocating the whitespace pre-floorplanning and 2) dynamically allocating that whitespace during floorplanning.

4.2.2.1 Whitespace allocation pre-floorplanning

Algorithm 1 shown below, is a method for preallocating whitespace to high power density blocks and is applicable to to both soft and hard blocks.

Algorithm 1 Pre-floorplanning Whitespace Allocation

Input:

Floorplanning Blocks

Output:

Floorplanning Blocks with size adjusted to reduce high power densities

1: Calculate available whitespace ($\mathrm{White_A}$)
2: **while** $\mathrm{White_A} > 0$ **do**
3: Get block with largest power density ($\mathrm{Block_{hp}}$)
4: Calculate enlargement area = Area of $\mathrm{Block_{hp}} \times \mathrm{A}_e$ (Area enlargement coefficient)
5: **if** enlargement area $>$ $\mathrm{White_A}$ **then**
6: enlargement area = $\mathrm{White_A}$
7: $\mathrm{White_A}$ = 0
8: Add enlargement area to $\mathrm{Block_{hp}}$
9: **else**
10: $\mathrm{White_A} = \mathrm{White_A}$ - enlargement area
11: Add enlargement area to $\mathrm{Block_{hp}}$
12: **end if**
13: **end while**

The first step of the algorithm consists of calculating the available whitespace for area utilization. The next steps of the algorithm is to find the block (Block_{hp}) with the largest power density, that can have whitespace allocated to it. Blocks that have been enlarged greater than 50% of their original area are not candidates to be enlarged since

enlarging a block by too great a factor significantly increases the wiring resources required to route within that block. The amount of whitespace to be added to Block$_{hp}$ is calculated as a percentage (Area enlargement coefficient - A$_e$) of the area of Block$_{hp}$. The value of A$_e$ affects the speed at which the algorithm finishes but also the quality of the final solution. A large value for A$_e$ ensures the whitespace is allocated quickly but runs the risks of unevenly distributing the whitespace amongst the blocks. A small value for A$_e$, requires a longer runtime but more evenly distributes the whitespace amongst the blocks. The final step of the algorithm adds the calculated amount of whitespace to Block$_{hp}$.

4.2.2.2 Whitespace allocation during floorplanning

The method for allocating whitespace during floorplanning adds additional moves to a simulated annealing floorplanner. For soft blocks the additional move is to randomly change the area of the block to some value that is between $1\times$ and $1.5\times$ it's original area. For hard blocks the additional move is to adjust the size of the virtual block around the hard block to be $1\times$ to $1.5\times$ of the hard block's original area.

4.3 Proposed Thermal Floorplanning

The proposed floorplanner uses a simulated annealing algorithm that is is similar to other floorplanners [17, 75]. The base simulated annealer has three moves: interchange two blocks by swapping both sequence pairs, displace a single block by swapping one pair in a single sequence pair, and the rotation of a single block. In addition to these normal moves, the two proposed perturbations that manage floorplan whitespace are added to the floorplanner. The cost function is changed to include the new power spreading cost in

addition to the standard maximum temperature cost like prior works [17, 75]. The final cost function for the floorplanner is

$$cost = \alpha \cdot \text{Area}_f + \beta \cdot \text{HPWL} + \eta T_{Max} + \lambda S_P \tag{4.4}$$

where T_max is the maximum chip temperature, S_p is the spreading cost and α, β, η, λ, are the different weights associated with each value. Each weight was selected empirically.

Chapter 5

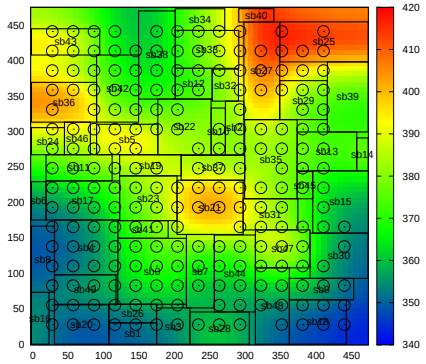
Mitigating C4 Bump Failures from Thermal-Cyclic Fatigue

The high temperatures found in modern ICs can lead to mechanical issues such a C4 bump failure from thermal-cyclic fatigue. Figure 5.1 shows two floorplans and bump placements for the same circuit, however the number of cycles to failure for the circuit in Figure 5.1(a) is 609 vs 82925 for the circuit in Figure 5.1(b). This chapter of the thesis discusses methods of floorplanning and bump placement co-optimization to increases the MTTF of C4 bumps from thermal-cyclic fatigue.

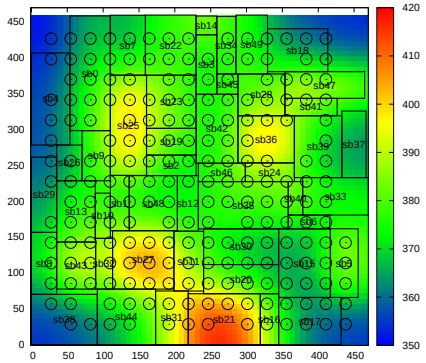
5.1 Bump Placement Effect on Package Reliability

The MTTF rate from thermal-cyclic fatigue for C4 bumps is strongly dependent on the magnitude of the thermal induced stress experienced by bump and the temperature of the bump.

The temperature of the bump affects the creep rate (Equation 3.29) which is di-



(a) n50 Floorplan 1: Min number of cycles to failure 609. Circuit dimensions in μm, temperature in Kelvin (K)



(b) n50 Floorplan 2: Min number of cycles to failure 82925. Circuit dimensions in μm, temperature in Kelvin (K)

Figure 5.1: Two Floorplans for the GSRC N50 Benchmark showing different bump placements and consequently different minimum number of cycles to failure.

rectly proportional to MTTF failure rate. Consequently, bumps should be placed to avoid chip hotpsots in order to increase their lifetime from thermal-cyclic fatigue.

The thermal-induced stress is linearly dependent on the distance of the bump from the centroid of the chip (Equation 3.25). Bumps that are close to the centroid have much less strain than those that are close to the chip edges as shown in Figure 5.2. It should be noted that the amount of strain is also influenced by the temperature (Equation 3.25).

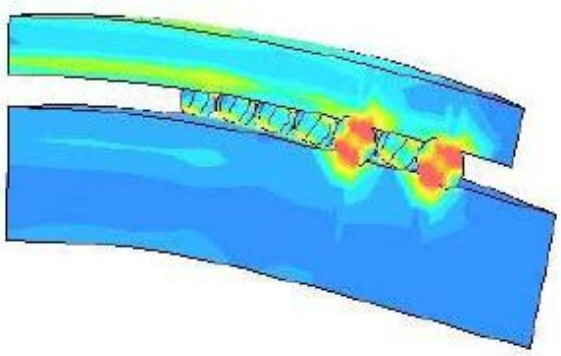


Figure 5.2: High Levels of thermal-induced stress in C4 bumps for a IC chip showing larger stress values closer to the edge of the chip. [83].

5.2 Floorplanning Effect on Package Reliability

The MTTF rate from thermal-cyclic fatigue for C4 bumps is strongly dependent on temperature (Equation 3.30). Chapter 4 shows that the temperature of a circuit can be lowered by the use of thermal-aware floorplanning. Consequently, using thermal-aware floorplanning is one method of increasing the MTTF of C4 bumps. Additionally, the MTTF is strongly dependent on the location of the temperature hotspots of the chips. Hotspots located at the edge of the chip lead to quicker MTTF rates than hotspots located toward the

centroid of the chip. Consequently, controlling the location of temperature hotspots during floorplanning can also lead to significant increases in MTTF rates for C4 bumps.

5.3 Floorplanning and Bump Placement Co-Optimization

The location of C4 bumps affects the floorplan and vice-versa, since one of the main goals of floorplanning and bump placement is to reduce the wirelength from the various blocks to bumps. Consequently, is important to co-optimize floorplanning and bump placement, since doing the bump placement before floorplanning or vice versa, greatly reduces the solution space that is searched. The proposed method of co-optimization consists of using a quadratic bump placement algorithm in conjunction with the thermal-aware floorplanner proposed in Chapter 4.

5.3.1 Quadratic Bump Placement

The proposed bump placement algorithm uses quadratic optimization to place the bumps in order to minimize wirelengths, then a greedy legalization procedure to remove bump overlaps. A detailed overview of the placement algorithm is shown in shown in Algorithm 2 and Figure 5.3.

The first step of the algorithm determines the optimal location of C4 bumps using quadratic wirelength optimization as shown in Figure 5.3(a). This consists of solving the following quadratic program

$$\min \frac{1}{2} \sum_{i}^{n} \sum_{j}^{m} w_{ij} \left(px_i - bx_j\right)^2 + \left(py_i - by_j\right)^2 \tag{5.1}$$

where w_{ij} is the weight of the connection between block j and bump i obtained from the

Algorithm 2 Quadratic Stress-Aware Bump Placement

Input:

Stress-Aware I/O Bump Placement.

Output:

All bumps satisfy a minimum number of cycles to failure.

1: Calculate optimal position for bumps via quadratic optimization.

2: Create a grid of possible bump locations.

3: Calculate the failure rate at all possible bump locations.

4: Prune possible bump locations.

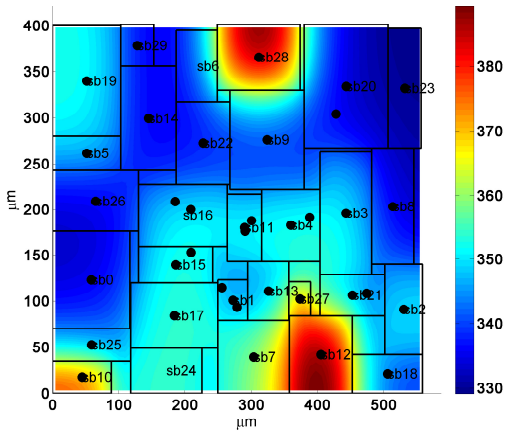
5: Greedy legalize bump positions.

device net-list, n is the number of bumps, m is the number of blocks, px_i and py_i are the x and y coordinates of bump i, and bx_i and by_i are the x and y coordinates of block j.

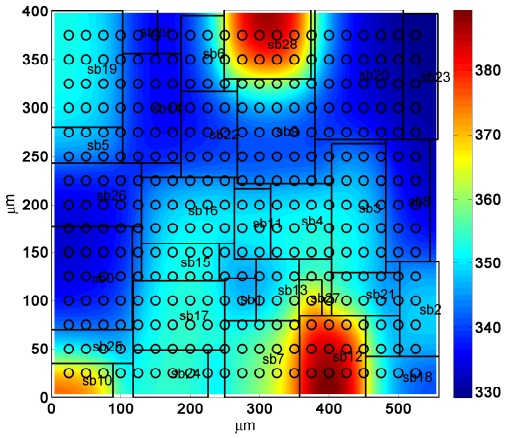
If a net in the device netlist only contains a single block and bump, then the weight of the connection between the block and bump is 1. However if the size of the net is greater than 2, then the clique model [106] is used to calculate the weight between each block and bump within the net.

After the optimal bump locations have been found the algorithm creates a grid of possible C4 bump locations using the minimum pitch distance between C4 bumps, and the length and width of the chip as shown in Figure 5.3(b).

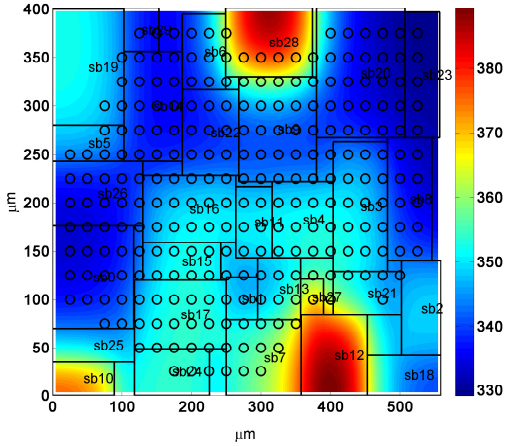
Once the grid has been created, the failure rate due to CTE mismatch between the substrate and die is calculated for every possible C4 location using a temperature map generated from the block layout and the fatigue models detailed in Section 3.3.5.3. The next step of the algorithm prunes all possible bump locations that have potentially low reliability as shown in Figure 5.3(c). There are two methods that can be used for pruning



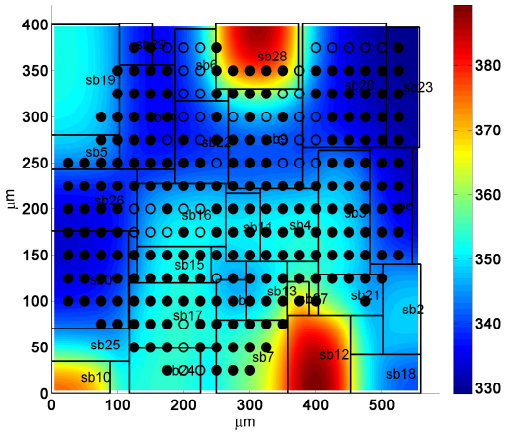
(a) Optimally place all C4 bumps (Temperature in Kelvin).



(b) Find all feasible bump locations (Temperature in Kelvin).



(c) Prune bump locations (Temperature in Kelvin).



(d) Legalize bump Locations (Temperature in Kelvin).

Figure 5.3: Example bump placement flow for n30 benchmark

bad bump locations. The first method specifies a minimum number of cycles to failure for all the bumps in the design. Given this value the algorithm will remove all possible bump locations with a value below that specified value for being a candidate location for a data C4 bump. The other method of pruning specifies a number of possible bump locations (N) that will not be used. The algorithm then removes the N possible bump locations with the lowest number of cycles to failure. The final step of the algorithm legalizes the bump placement from the first step as shown in Figure 5.3(d). It should be noted that the number of feasible bump locations

5.3.1.1 Bump Legalization

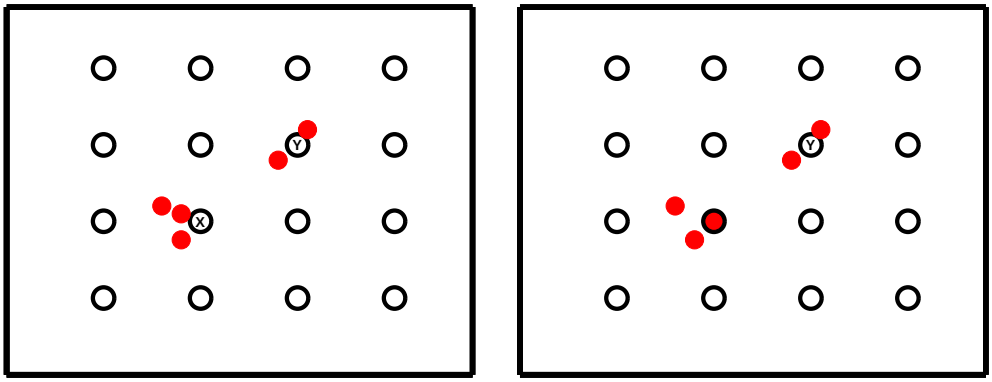
Legalization assigns a bump to one of the possible bump locations on the grid array calculated in stage 2 minus the locations removed from stage 4 of the algorithm. The legalization employed by the proposed algorithm is a 2 step greedy procedure. In the first step, for each possible bump location the number of bumps closest to that location is tabulated and placed in a bin. The possible bump location bins are then sorted so that the bin with the largest number of bumps is first in the list. Finally, for each bin in the list, the bumps are greedily place as close to its optimal location avoiding overlaps with other bumps and the locations that have been removed for reliability issues. An example flow for the legalization is shown in Figure 5.4.

Figure 5.4(a) shows the position of the bumps in illegal positions after the quadratic wirelength optimization step. The first step is to create bins for each bump location. For this example bump location x has 3 bumps located nearby and bump location y has 2 bumps located nearby. The next step of the legalization process is to legalize the positions of

the bumps within each bin starting with the largest bin, which is bump location x. The closest illegal bump to a legal bump position is placed in that bump position as shown in Figure 5.4(b). The other bumps within that bin are then legalized to the other bump locations closest to them that do not have a bump already placed there as shown in Figures 5.4(c) and 5.4(d). Once all the bumps within a bin are placed, the procedure moves to the next largest bin as shown in Figures 5.4(e) and 5.4(f). The procedure is repeated until all the illegal places bumps have been placed in legal bump positions.

5.3.2 Proposed Floorplanning and bump placement Co-optimization

The proposed floorplanning and bump-placement co-optimization uses the thermal floorplanner described in Section 4.3 and the quadratic bump placement algorithm detailed in Section 5.3.1. The floorplanner has the same cost function as the one described in Section 5.3.1, however, there is one additional move which is a call to the bump placement algorithm. The bump placement algorithm is called at a much lower frequency than the other moves, such as swapping blocks since the runtime for the bump placement is significantly larger than the other moves in the floorplanner. Consequently more frequent calls to the bump placing algorithm would lead to significantly longer execution times.



(a) Initial illegal placement

(b) Legalize first bump

(c) Legalize second bump

(d) Legalize third bump

(e) Legalize fourth bump

(f) Legalize final bump

Figure 5.4: Example bump placement flow for n30 benchmark

Chapter 6

Reducing High Temperatures in Interconnect

Another reliability issue caused by high on chip temperatures in addition to thermal-cyclic fatigue of solder balls is increased interconnect electromigration and increased interconnect resistance. These issued are mitigated by reducing interconnect temperatures. Interconnect temperatures are reduced by decreasing the total amount of Joule heating in the interconnect. This chapter of the thesis discusses a methodology to reduce high temperatures in the Power Supply Network (PSN) interconnect by redistributing decoupling decapacitance throughout the chip.

6.1 Using Decoupling Capacitance to Reduce PSN Interconnect Temperatures

One method of decreasing high temperatures in PSN interconnect is decreasing the RMS current in the interconnect and the distance that the current has to travel through the interconnect. Most power is supplied to a circuit from the package bumps/pins, but unfortunately, this power must go through the various metal interconnect layers before reaching the transistors on the substrate. This can lead to a significant amount of energy loss due to Joule heating through the PSN wires.

Decoupling capacitance (decap) also provide power to circuits, but are located electrically closer to the transistors than the power supply bumps (Section 2.6). Consequently, placing decaps in areas that require significant power will reduce the Joule heating in the interconnect by reducing the distance in wires that the power supply current has to travel and also reducing the current spikes in the wires.

Decap is thus a viable mechanism for reducing Joule heating in interconnect since it reduces the Joule heating RMS value. Decap is usually added to a design to reduce transient voltage droop (Section 2.8.3), however, this decap placement is usually quite flexible with multiple placements meeting the voltage droop requirement. Some decap can be redistributed to reduce Joule heating while still ensuring that the voltage droop requirements are met. Additional decap is added in areas that have high Joule heating while decap is removed from areas that do not have significant Joule heating and are well within the voltage droop bounds in order to preserve PSN integrity. Figure 6.1 shows an example where redistributed decap was able to reduce the Joule heating power in the PSN interconnect for

an IBM powergrid benchmark.

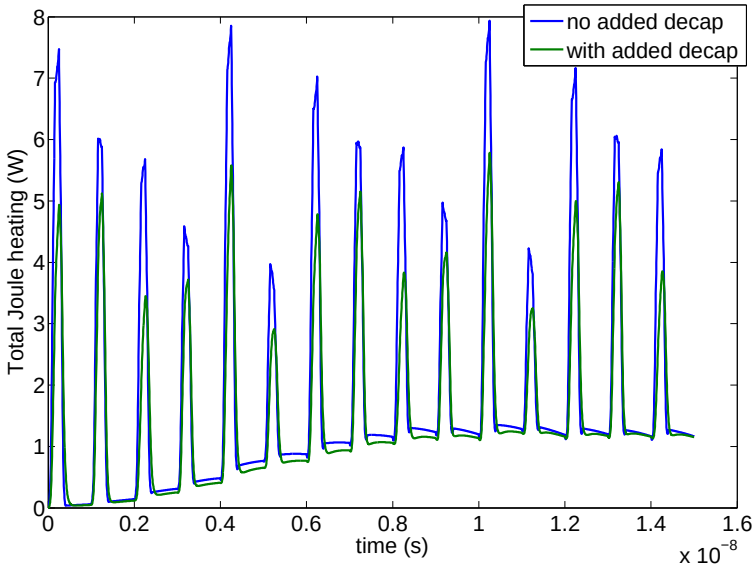


Figure 6.1: Decap is a viable mechanism for reducing Joule heating power as can be seen by the reduced peaks of total Joule heating.

6.2 Decap Redistribution Algorithm to Reduce Joule Heating

The proposed method of decap redistribution uses a gradient descent, non-linear optimization. The design is spatially partitioned so that the sensitivity to reduce Joule heating of each partition with respect to the decap value can be measured. This enables decap to be shifted from less sensitive partitions to the more sensitive partitions. A detailed implementation of the proposed redistribution algorithm is presented in Algorithm 3.

6.2.1 Partitioning and Budgeting

The first stage of the algorithm calculates the total decap redistribution budget, C_{budget}. The design is partitioned into various regions at the block level or potentially a

finer granularity. Next, each partition of the PSN is simulated with a small, fixed amount of decap removed from that partition to determine if removing decap causes a voltage droop violation. If so, that partition is flagged so that decap can only be added to it and not removed. If removing decap does not cause a violating voltage droop in the partition, the decap is permanently removed and added as surplus to the overall decap redistribution budget, C_{budget}. The amount of decap removed during each simulation, C_{rem}, is a variable within the algorithm. Large values for C_{rem} limit the number of partitions that contribute to C_{budget} since removing a large amount of decap will likely cause excessive voltage droop. On the other hand, small values of C_{rem} limit the effectives of the algorithm due to the small amounts of decap added to the redistribution budget. Finally, C_{budget} is then uniformly distributed to the redistribution decap budget (δC_i) for each partition as an initial redistribution which is subsequently improved on. The initial decap in each partition's δC_i is consequently $\frac{C_{\mathrm{budget}}}{N}$ where N is the number of partitions.

6.2.2 Sensitivity Simulation

The refinement stage begins by calculating the sensitivity of the total Joule heating in each partition to the decap in that partition, $\frac{dJ_i}{dC_i}$. This determines which partitions will most (or least) benefit from additional decap. The sensitivities are measured by simulating the entire PSN with the present decap allocation and calculating the total Joule heat, J_i^{part1}, within each partition i. The PSN is then re-simulated with each partition containing a small amount of added decap, δ_{add}, to calculate a forward finite difference. The total Joule heating for each partition with the added decap is calculated as J_i^{part2}. The average decap

Algorithm 3 Decap Redistribution to Minimize Joule Heating

Input: Decap placement satisfying voltage droop requirements.

Output: Decap placement minimizing total Joule heating while still satisfying voltage droop requirements.

1: Partition design into N partitions
2: Determine total redistribution budget (C_{budget})
3: Uniformly allocate C_{budget} to each partition's redistribution budget (δC_i)
4: **repeat**
5: Calculate sensitives ($\frac{dJ_i}{dC_i}$) for each partition
6: Calculate mean of sensitivities
7: Find low sensitivity partitions (sensitivity value below mean)
8: Find high sensitivity partitions (sensitivity value above mean)
9: Redistribute decap from the low sensitivity partitions δC_i to the high sensitivity partitions δC_i
10: Calculate change in total Joule heating, ΔJ_{total}
11: **until** $\Delta J_{\text{total}} \leq \epsilon$

sensitive for a partition can thus be calculated as

$$\frac{dJ_i}{dC_i} = \frac{J_i^{\text{part2}} - J_i^{\text{part1}}}{\delta_{add} N_{wire}} \tag{6.1}$$

where N_{wire} refers to the number of wires within the partition and C_i refers to the decap in partition i. The decap sensitivity is normalized by the number of wires since some partitions contain more wires than others and the sensitivities calculated would be skewed to those partitions if they were not normalized.

6.2.3 Reallocation

The final stage of the algorithm determines how to incrementally redistribute the decap, C_{budget}, across all the partitions. First, the mean of the partition sensitivities is computed. All partitions with sensitivities above the mean are flagged as partitions to receive more decap and those partitions below the mean lose decap. The amount of decap that is removed or added to a partition is based on the magnitude of its sensitivity.

Partitions with sensitivities below the mean lose the following percentage of their redistributed decap from their δC_i

$$\text{Scale}_i = \frac{\zeta \left(S_{\text{mean}} - S_i\right)}{S_{\text{mean}} - \min(S_{\text{low}})} \tag{6.2}$$

where Scale_i is the percentage of redistributed decap to remove from the partition, S_{mean} is the mean of the sensitivities, S_i is the sensitivity of partition i, ζ is the maximum fraction of decap that can be redistributed from a single partition at each iteration and S_{low} is a set containing all the sensitivities below the mean.

Partitions with sensitivities above the mean gain the following percentage of the redistributed decap removed from the lower sensitivity partitions to their δC_i:

$$\text{Scale}_i = \frac{S_i}{\sum S_{\text{high}}} \tag{6.3}$$

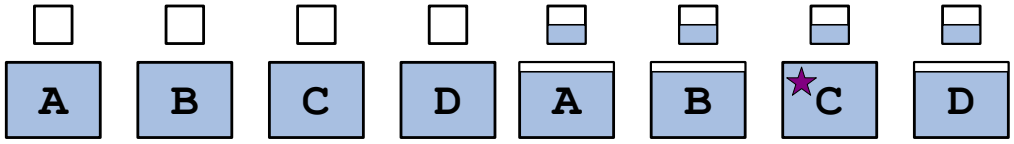
where S_{high} is a set containing all the sensitivities above the mean.

At each iteration of the algorithm only a fraction (controlled by the variable ζ) of decap from the δC_is is redistributed. A ζ of one leads to the algorithm finishing in one iteration, however the redistribution is only based on the initial sensitivities and consequently such a solution might be far from optimal. Incrementally redistributing the decap obtains

better solutions since as the partitions with large sensitivities get more decap their sensitives diminish and other partitions that did not have a high sensitivity become the leading candidate for redistributed decap. The redistribution process is repeated until the change in total Joule heating (J_{total}) between iterations is below a threshold or the max number of iterations allowed is reached.

6.3 Algorithm Example

The algorithm is further illustrated using a simple example. The first stage of the algorithm consists of partitioning the design into blocks (Figure 6.2(a)) and then calculating the C_{budget} and distributing it evenly to all the blocks (Figure 6.2(b)). In Figure 6.2(b), C_{budget} can be interpreted as the sum of all the decap in the δC_is.



(a) Initial circuit is partitioned into 4 blocks A-D. The smaller box above each image represents δC_i for that block.

(b) Decap removed from blocks that do not cause voltage droop is redistributed evenly to the δC_is for all blocks. Note that block C is flagged (Purple star) meaning no decap can be removed from that block.

Figure 6.2: Initial decap allocation and redistribution for first stage of the algorithm with decap percentages represented by the size of the shaded area.

Next, the Joule heating sensitivity with respect to the redistributed decap is calculated for each block which, for this example, is set to $\{0.1, 0.4, 0.7, 0.8\}$. Blocks A and

B are then flagged as low sensitivity blocks (mean sensitivity is 0.5) while Blocks C and D are flagged as high sensitivity blocks.

Finally, the C_{budget} is redistributed from the δC_is of the low sensitivity blocks to the δC_is of the high sensitivity blocks as illustrated in Figure 6.3. The amount of redistributed decap removed/added to each block depends on the sensitivity. Block A, has the lowest sensitivity hence the percentage of decap redistributed from Block A is the maximum (ζ) while Block B only loses $\zeta \times \frac{0.5-0.3}{0.5-0.1}$. Block C gains $\frac{0.7}{0.7+0.8}$ of the decap redistributed from Blocks A and B while Block D gains $\frac{0.8}{0.7+0.8}$ of the redistributed decap. The redistribution process is repeated until the stopping criterion is reached.

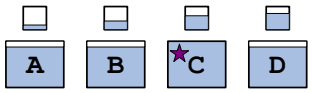


Figure 6.3: Decap is moved from the δC_is of the low sensitivity blocks (A and B) to the δC_is of the high sensitivity blocks (C and D)

6.4 Decap Redistribution Algorithm to Reduce Temperature

The temperature increase in the interconnect is directly proportional to the Joule heating RMS power and thermal resistance of the interconnect to the substrate. Consequently, for maximum reduction in interconnect temperatures, the thermal resistance should also be incorporated during the decap redistribution. Reducing the Joule heating in a large wire with a small thermal resistance is not as effective at increasing reliability as reducing Joule heating in a large wire with a large thermal resistance and significant Joule heating. Thus, a second version of Algorithm 3 is implemented to directly reduce interconnect temperature as opposed to reducing Joule heating.

To account for interconnect temperatures, the sensitivity used for the gradient-based decap redistribution sensitivity on Line 3 is instead calculated with respect to ΔT, the maximum interconnect temperature increase due to Joule heating as opposed to the total Joule heating power. This new sensitivity is calculated using

$$\frac{d\Delta T_i}{dC_i} = \frac{\Delta T_{\text{part2}_i} - \Delta T_{\text{part1}_i}}{\delta_{add}} \tag{6.4}$$

where $\Delta T_{\text{part1}_i}$ and $\Delta T_{\text{part2}_i}$ are the maximum wire temperature increase in partition i under the present decap allocation and with an additional δ_{add} decap, respectively. As opposed to the Joule heating sensitivity, this is not normalized to the number of wires since it is the maximum temperature in the partition. The maximum wire temperature increase is used for the optimization as opposed to the average ΔT, since reducing maximum temperatures is more important for interconnect reliability.

Finally, Line 3 of the algorithm is adjusted to calculate the change in Max ΔT and the stopping criterion in Line 3 is adjusted to consider change in Max ΔT instead of ΔJ_{total}.

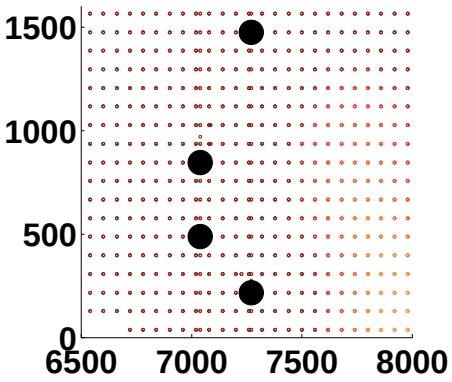
Chapter 7

Redundant Power Bump Placement

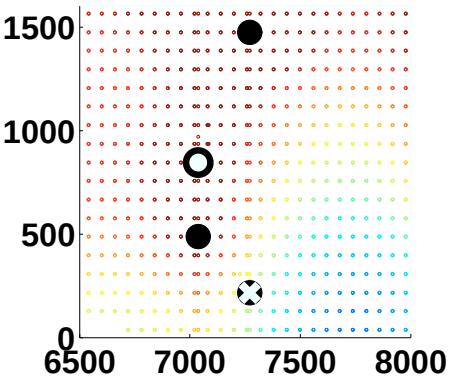
Large chip temperatures increase the rate of electromigration in power supply bumps leading to bump defects (Section 3.3.4). Bump defects can also be created from thermal-cyclic fatigue as detailed in Chapter 5 and also manufacturing defects. Recently, there has been an increase in manufacturing bump defects which has been exacerbated by the increase in bump counts and shit to no-flow underfill processes [87]. and also yield issues. Specifically, no-flow underfill processes are susceptible to the void formation, non-wetting of solders and chip floating [41].

The formation of a defect in a C4 bump used in the PSN can cause static and transient voltage violations in the power grid as shown in Figure 7.1(b). In addition, a bump defect in a power supply bump can also cause electromigration failures in other power supply bumps as shown in Figure 7.1(b).

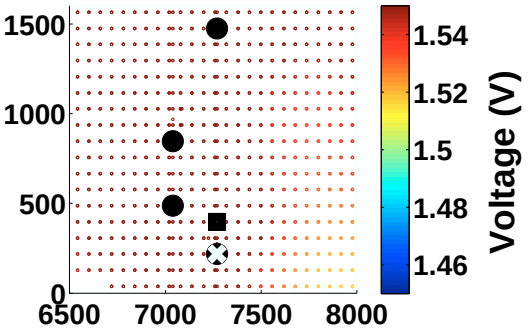
This chapter discusses methods of placing redundant power bumps within a circuit, so that in case of any single bump failure the power supply network will still meet the voltage and electromigration constraints as shown in Figure 7.1(c).



(a) Voltage map corresponding to all functional bumps. Dimensions in μm.



(b) Static voltage violations and MTTF failures (hollow bump) caused by a single-bump defect (white cross). Dimensions in μm.



(c) Redundant bump (square) removes static voltage violations and electro-migration failures caused by single-bump defect. Dimensions in μm.

Figure 7.1: Example from ibmpg2 benchmark showing voltage violations and electro-migration failures caused by a single-bump defect. Bumps are represented as large circles and node voltages represented as small circles. Only a small fraction of the entire benchmark is shown for image clarity with dimensions in μm.

7.1 Bump Failure Classification

As stated in the Chapter introduction bump failures in Power supply bumps can lead to reliability issues in the Power Supply Network. One bump failure may lead to transient/static voltage failures and/or electromigration failure in other bumps. Note, however that not all bump failures will cause reliability issues. Bump failures can be classified by the type of failure they induced in the PSN: voltage violations and/or electromigration violations.

7.1.1 Voltage violations

The bump failures which cause voltage violations in the PSN, belong to the voltage failure bump set, V_p. The bump that fails in Figure 7.1(b) belongs to V_p since it causes static voltage violations upon failure. Note, however, that not all the bumps in a design will cause static voltage violations if they have a bump defect since there is inherent redundancy in a power grid architecture.

7.1.2 Electromigration violations

The bump failures which cause an electromigration failure in neighbouring bumps, belong to the electromigration failure critical bump set (C_p). Electromigration failure in neighborhood bumps occur because of the increased current load on those neighboring bumps. The bumps that fail due to a defect/failure of a bump in C_p belong to the electromigration failing bump set (F_p).

Figure 7.2 further shows the bipartite many-to-one relationship between the F_p and C_p sets. Each bump in F_p corresponds to one or more bumps in C_p since multiple

bumps in C_p might cause the same bump failure in F_p. The C_p, F_p and V_p sets are not mutually exclusive and it is possible for a bump to belong to multiple sets as shown in Figure 7.2 and the example used in Figure 7.1.

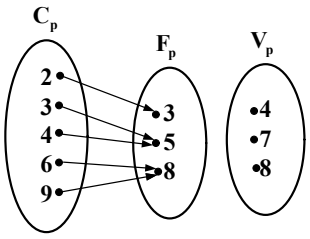


Figure 7.2: Figure showing the relationship between the F_p, C_p and V_p the sets.

7.1.3 Calculating Bump Failure Violations

The C_p, F_p and V_p sets are determined by doing N static PSN simulations with a different bump being removed during each simulation, where N represents the number of power bumps in the design. For each simulation, the current through each bump is checked to ensure that it is less than the maximum current threshold (I_{thres}) for that bump location including the effect of intra-die temperature. The I_{thres} for each possible bump location is calculated using Equation 3.23 and the temperature map of the circuit. The static IR voltage drop is checked to ensure that it is above some specified threshold value. Such a simulation is an upper bound on the estimation of F_p and V_p since often a bump merely has an increase in resistance due to a manufacturing or electromigration issue but not a complete failure.

7.2 Bump Redundancy

Bump redundancy is a possible method to fix power supply network failures due to a single-bump defect/failure. Figure 7.1(c) shows how a redundant bump (square) was able to fix the voltage violations and electromigration failures created by a single bump failure. Many designs in practice guard-band by inserting extra bumps but not in a formal manner. This section outlines methods for creating redundant bumps so that the design is guaranteed single bump redundant.

There are two major challenges with generating a redundant bump set (R_p) for single-bump robustness, namely, selecting which bumps need to be made redundant, and determining which additional bumps can provide coverage. The bumps to be made redundant are selected from the F_p, C_p, and V_p sets to form an overall critical bump set (O_p). The coverage for each possible redundant bump location is calculated and then tabulated as a set p_{cov_i} that contains the bumps from the O_p set that are covered by redundant bump i.

The possible redundant bump locations are selected from the top level nodes of the PSN that are not located within the minimum bump spacing requirement of other bumps in the circuit. The minimum bump space requirement is required for manufacturing purposes. A summary of the different set definitions are given in Table 7.1.

7.2.1 Generating the Redundancy Coverage Sets

Adding a redundant bump to a design will affect the power supply network within a certain distance of the redundant bump due to the locality effect [13]. The effectiveness of adding a redundant bump in removing failures caused by single bump defect, however,

Table 7.1: Definitions of Terminology

Set	Definition
V_p	Set of bumps that cause static voltage violations if they have a defect/failure.
C_p	Set of bumps that cause electromigration failures in other bumps if they have a defect/failure.
F_p	Set of bumps that will have an electromigration failure due to a defect/failure of a bump in C_p
R_p	Set of redundant bumps added to design for robustness.
O_p	Set of bumps from the F_p, C_p, and V_p sets to be made redundant
p_{cov_i}	Set of bumps from the O_p set that are covered by redundant bump i

depends on the resistivity of the power grid in that area of the PSN, the size and number of current sources in that area of the PSN, the number of surrounding bumps, and the size and magnitude of the failures caused by the single-bump defect. Since these factors vary throughout a design, the effectiveness of a redundant bump has to be empirically estimated for each individual bump defect.

7.2.1.1 Electromigration Coverage Sets

Redundant bump coverage for bumps in F_p entails removing or reducing the extra current in the bump as a result of single bump failure in C_p. We define the current slack in a bump as

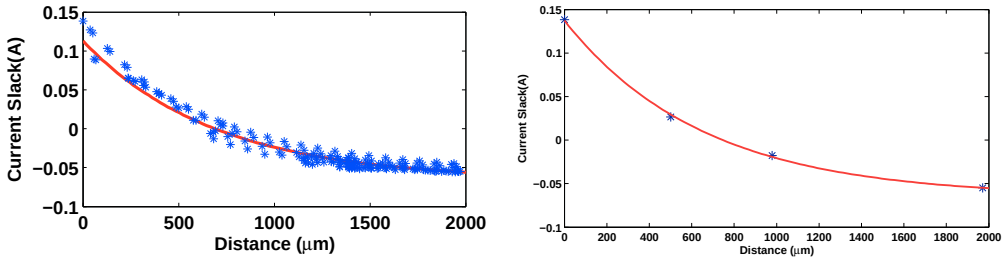
$$I_{slack} = I - I_{thres} \tag{7.1}$$

where I is the current through the bump and I_{thres} is the maximum allowable current through that bump. Bumps in F_p can have a negative current slack depending on what member of the C_p sets fails. A redundant bump covers a bump in F_p to ensure that the I_{slack} always remains positive irregardless of which bump from the C_p sets fails. Consequently, calculating the coverage set for a bump in the F_p set requires two steps.

The largest negative slack for bumps are determined during the generation of the F_p, C_p and V_p sets. The slack values for each bump in F_p are calculated for the different bumps failures from the C_p set and the maximum negative slacks are used for coverage set generation. Not using the maximum negative slack will overestimate the coverage set for that bump.

The relationship between the current slack in the F_p bump and the distance of

an added redundant bump was empirically observed to be exponential as shown in Figure 7.3(a), but depends on several factors such as number of close by bumps, the resistivity of the underlying mesh in that area, and the total current draw in that area. The relationship



(a) Change in current slack in failing bump vs distance of added bump demonstrating exponential relationship

(b) Four random points are used to approximate the exponential relationship without significant loss in accuracy while reducing run-time overhead

Figure 7.3: Current slack model showing exponential relationship and approximated relationship with fewer data points.

between current slack and distance of an added bump is fitted using

$$I_{slack} = A \exp\left(-B \cdot p_{dist}\right) + C \tag{7.2}$$

where A and B and C are constants based on the current draw and the resistivity of the mesh and p_{dist} is the distance of the added bump to the failing bump. An example of this phenomenon for the ibmpg2 benchmark is shown in Figure 7.3(a) as the least squares fitted curve to a linearized form of Equation 7.2.

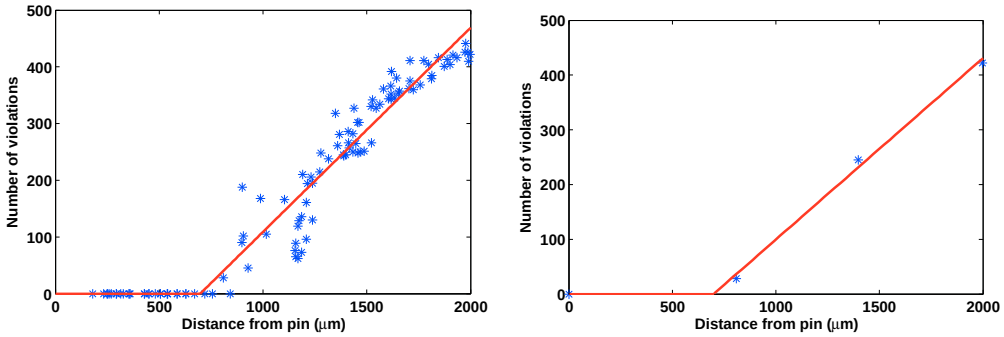
It is not practical to simulate the effect on the slack for each candidate bump location, consequently, for our experiments the current slack vs distance graph was constructed from 4 distances of 100, 500, 1000 and 2000 μm. These distances were chosen to capture

enough information about the relationship to obtain a good approximation for the A and B and C constants. More data points would lead to a significant increase in runtime due to the number of simulations required, but four data points is a good trade-off between accuracy and runtime.

The discrete nature of the grid limits exact distances so the closest candidate bump location were used. Figure 7.3(b) shows the graph using these four points. The root of the equation generated with the reduced set of data points only differs from the root of the equation with all data points by 7%.

The coverage set for each bump in F_p set is calculated using Equation 7.2. The root of Equation 7.2 corresponds to a distance value at which the slack of the failing bump becomes zero, and consequently any bump placed at a candidate bump location with a distance (d_{cover}) less than or equal to that distance will ensure there is no MTTF violations. All such candidate bumps comprise the coverage set for a failing bump. A redundant bump in the coverage set of a bump in F_p provides complete redundant coverage for that bump without needing help from other bumps.

However, candidate redundant bumps that lie beyond d_{cover} from the failing bump still reduce the slack in the failing bump, but not completely. These bumps belong to the partial coverage set for that failing bump. A redundant bump in the partial coverage set of a failing bump cannot completely provide redundant coverage for that failing bump, but in conjunction with other bumps, could provide complete coverage. Partial coverage allows a larger solution space and usually in fewer redundant bumps. The partial coverage value for a candidate redundant bump is determined by calculating the slack removed divided by the total negative slack for that failing bump using estimates from Equation 7.2.



(a) Change in number of voltage violations vs distance of added bump demonstrating linear relationship

(b) Four random points are used to approximate the linear relationship without significant loss in accuracy while reducing run-time overhead

Figure 7.4: Linear model of redundant bump placement for a failing bump causing static voltage failures in the ibmpg2 benchmark

7.2.1.2 Voltage Coverage Sets

The coverage for a bump in V_p is calculated similarly to a bump in F_p and is based on a distance based metric. If a bump in V_p fails, nodes within close proximity will have static voltage drop violations. Adding a bump at a location close enough to the bump from V_p will remove these violations. The distance of the added bump is also empirically estimated and was observed to be linear. Consequently, the relationship was modelled as

$$V_n = M \cdot p_{dist} + D \tag{7.3}$$

where M and D are constants and V_n is the number of node voltage violations. An example from the ibmpg2 benchmark depicted in Figure 7.4(a) illustrates the relationship between the number of static voltage violations when a single bump from V_p is removed and the distance of an added redundant bump. After a certain distance, d_{cover} there are no more

voltage violations. Consequently, the potential redundant bumps located within the d_{cover} of a voltage failing bump comprises the redundancy set for that bump. Similar to the MTTF coverage set generation, simulating all the potential bump locations for each voltage bump failure is too time consuming. Consequently, four data points capture the relationship between voltage violations and distance of the added bump. The distances chosen for are 200, 800, 1400 and 2000 μm. The fitted graph for the example shown in Figure 7.4(a) using the reduced set of data points is shown in Figure 7.4(b) and has an error in the root of the equation of about 10% for this example which is a typical error value.

7.3 Redundant Bump Set Generation

Single-bump redundancy augments a power bump placement with a minimal redundancy bump set (R_p) such that for any single-bump defect, nearby bumps do not suffer from a cascading electromigration failure and all nodes within the design continue to satisfy minimum static and transient voltage constraints.

The C_p, F_p and V_p sets guide the redundant bump placement since they identify which bumps need to be made redundant. Consequently, R_p set generation finds the mapping of redundant bumps locations to bumps in the various failure sets to minimize the number of redundant bumps required. Since there are no prior works in this regards, three alternative methods are proposed.

7.3.1 Naive Bump Redundancy

The Naive method uses a simple greedy algorithm as detailed in Algorithm 4. First, the C_p and V_p sets are computed using the initial bump placement and then the coverage sets for both the C_p and V_p sets are generated. The O_p set is then computed based on the size of C_p and F_p. If $|C_p| < |F_p|$ then $O_p = C_p \cup V_p$ otherwise $O_p = F_p \cup C_p$. Finally, R_p is computed by visiting every bump in O_p and adding the closest redundant bump to R_p.

Algorithm 4 Naive Algorithm

Input: Original bump placement

Output: Set of redundant bumps locations R_p

1: Generate C_p, F_p, V_p and O_p sets.

2: **repeat**

3: Select bump j in O_p

4: Select redundant bump closest to j and add to R_p

5: **until** All bump in O_p are visited

7.3.2 Improved Greedy Bump Redundancy

The Improved Greedy Algorithm greedily adds bump with the largest coverage sets to R_p as detailed in Algorithm 5. First, the various redundant sets are calculated in a similar fashion as for the Naive Greedy method. Then, R_p is computed by sorting all the p_{cov_i} sets by size and selecting the redundant bump with the largest p_{cov_i} to add to R_p. All bump from O_p covered by that redundant bump are removed from the p_{cov_i} of the other redundant bumps and the process is repeated until all bumps from (O_p) are covered.

Algorithm 5 Improved Greedy Algorithm

Input: Original bump placement

Output: Set of redundant bump locations R_p

1: Generate C_p, F_p, V_p, p_{cov_i} and O_p sets.

2: **repeat**

3: Sort p_{cov_i} by size

4: Add candidate redundant bump with largest p_{cov_i} to R_p

5: Adjust remaining p_{cov_i} sets.

6: **until** All bump in O_p are covered

7.3.3 ILP Bump Redundancy

Most possible redundant bump locations can cover multiple bumps from the F_p, V_p and C_p sets. Consequently, if the right mix of redundant bumps are selected the total size of the R_p can be reduced significantly.

Selecting a minimal R_p based on bumps in O_p is a set covering problem which is solvable as an ILP. Given the possible redundant bump locations and the bumps to be covered in O_p the ILP picks the least number of redundant bumps so that every bump within O_p is covered. More formally the ILP formulation is

$$\min \quad \textstyle\sum_i r_i,\ r_i \in \{0,1\}$$

$$\text{s.t.} \quad A_f \cdot r \geq \mathbf{1}$$

where r is a binary vector for all possible redundant bump locations and specifies whether a bump is placed at that location or not, A_f is a matrix that contains the coverage information for all possible redundant bump locations obtained from the coverage sets P_{cov}. A_f is a $M \times N$ matrix where N is the number of possible redundant bump locations and M is the size of the O_p set to be covered. Column i in A_f contains the bumps from O_p that are

covered by placing a redundant bump at position i. The ILP solver will choose the smallest R_p set to ensure all the bumps in the O_p set are covered.

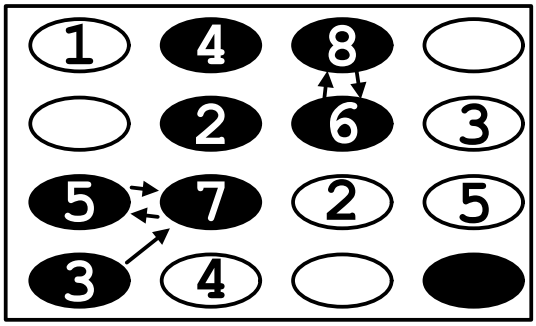
Another advantage of an ILP formulation is to incorporate partial coverage as detailed in Section 7.2.1.1. Two or more potential redundant bumps might provide redundant coverage for a bump even though individually those bumps cannot provide full coverage for that bump. Considering partial coverage only requires changing the A_f matrix to include fractions. For example, $A_{f_{i,j}} = k\cdot$, where k is a fraction between 0 and 1, implies that redundant bump location j covers O_i partial by a value of k.

7.4 Algorithm Example

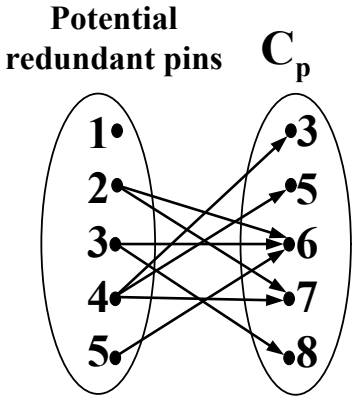
The proposed algorithms are further illustrated using a simple example. Figure 7.5(a) shows a simple bump layout for a example circuit with existing bumps represented as solid circles and potential redundant locations as hollow circles. For this example V_p is $\{2, 4, 5, 6, 8\}$, F_p is $\{5, 6, 7, 8\}$ and C_p is $\{3, 5, 6, 7, 8\}$. The coverage for each redundant bump for the bumps in C_p, F_p and V_p are illustrated in Figures 7.5(b), 7.5(c) and 7.5(d) respectively.

7.4.1 Calculating O_p and the P_{cov}

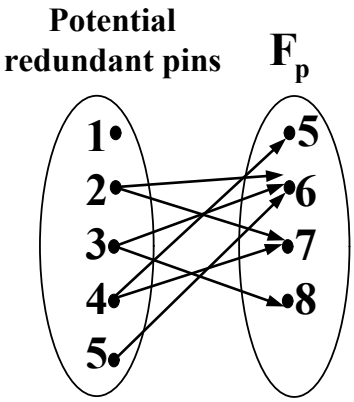
The first step in all algorithms is determining the O_p set. Since $|C_p| > |F_p|$ then $O_p = F_p \cup C_p$ as shown in Figure 7.6 (O_p = $\{2, 4, 5, 6, 7, 8\}$,). Once O_p is calculated for the circuit the next step is to determine the coverage for each possible redundant bump location. From Figure 7.6 the redundancy coverage sets (P_{cov}) for each possible redundant bump is



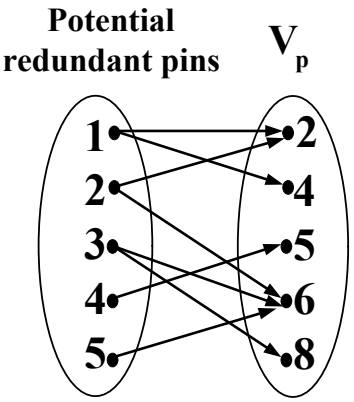
(a) Bump placement showing original bump (solid) and potential redundant bump locations (hollow). Arrows show the bump in F_p that fail from bump in C_p



(b) Relationship between potential redundant bump and bump in C_p



(c) Relationship between potential redundant bump and bump in F_p



(d) Relationship between potential redundant bump and bump in V_p

Figure 7.5: Example bump placement, showing corresponding redundancy mappings. An arrow from a the redundant bump location to a bump in the opposing set, means that bump is covered by that redundant bump location.

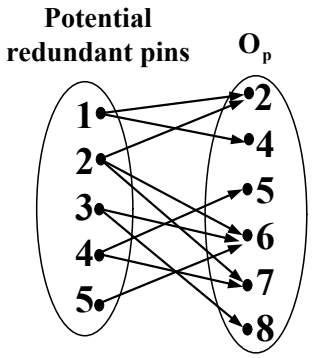


Figure 7.6: Relationship between potential redundant bump and bump in O_p ($F_p \cup V_p$)

as follows:

p_{cov_1} = $\{2, 4\}$

p_{cov_2} = $\{2, 6, 7\}$

p_{cov_3} = $\{6, 8\}$

p_{cov_4} = $\{5, 7\}$

p_{cov_5} = $\{6\}$.

7.4.2 Naive

The Naive algorithm first computes O_p and then iterates through each member selecting a redundant bump to provide coverage. Thus the algorithm selects redundant bump 1 to cover bumps 2 and 4, then redundant bump 4 to cover bumps 5 and 7, etc., until all bumps in O_p are covered. The final R_p is thus $\{1, 4, 5, 3\}$.

7.4.3 Improved Greedy

The Improved Greedy algorithm first computes O_p like the Naive algorithm. Then the coverage sets for each possible redundant bump location is sorted by size as follows:

1) p_{cov_2} = $\{2, 6, 7\}$

2) p_{cov_1} = $\{2, 4\}$

3) p_{cov_3} = $\{6, 8\}$

4) p_{cov_4} = $\{5, 7\}$

5) p_{cov_5} = $\{6\}$.

The next step is selecting the redundant bump with the largest coverage set and adding it to R_p which in this case would be redundant bump 2. The bumps covered by that redundant bump are then removed from the remaining coverage sets since they are already covered as follows:

1) p_{cov_1} = $\{4\}$

2) p_{cov_3} = $\{8\}$

3) p_{cov_4} = $\{5\}$

4) p_{cov_5} = $\{\}$.

The process is repeated until all bump in O_p are covered leading to a final R_p of $\{2, 1, 3, 4\}$.

7.4.4 ILP

The first step of using the ILP formulation is generating the A_f matrix from the various coverage sets for each potential redundant bump location. First the constraint equations for each bump in O_p with relation to the potential redundant bump is created as shown in Figure 7.7(a). Then the A_f matrix is created from these constraints as shown in Figure 7.7(b)

After the A_f matrix is generated, it is passed to an ILP solver to choose the smallest R_p to ensure all the bumps in O_p are covered, which in this case is $\{1, 4, 3\}$, which is smaller than the R_p from the Naive and Improved Greedy methods.

7.4.4.1 Partial Coverage

Another advantage of an ILP formulation is to incorporate partial coverage as detailed in Section 7.3.3. Two or more potential redundant pins might provide redundant coverage for a pin even though individually those pins cannot provide full coverage for that pin. For example, lets assume the maximum current slack for pin 8 and pin 5 is 0.2A, and that redundant pins 1 and 2 can each cover 0.1A of that slack. Individually, redundant pins 1 and 2 would not be able to fully cover pins 5 and 8, but together they could since redundant pin location 1 would provide 50% coverage and redundant pin location 2 would provide the other 50% coverage for pins 5 and 8. Incorporating that specific partial coverage scenario results in new constraints equations (Figure 7.8(a)) and A_f matrix (Figure 7.8(b)). The ILP solution is is reduced to $\{1, 2\}$ since those two bumps together provide redundant coverage to all the bumps in O_p.

$$
\begin{aligned}
r_1 + r_2 &\geq 1 \quad (O_2) \\
r_1 &\geq 1 \quad (O_4) \\
r_4 &\geq 1 \quad (O_5) \\
r_2 + r_3 + r_5 &\geq 1 \quad (O_6) \\
r_2 + r_4 &\geq 1 \quad (O_7) \\
r_3 &\geq 1 \quad (O_8)
\end{aligned}
$$

(a) ILP constraint equations for each bump in O_p

$$
A_f = \begin{array}{c} \\ O_2 \\ O_4 \\ O_5 \\ O_6 \\ O_7 \\ O_8 \end{array}
\begin{array}{c}
\begin{array}{ccccc} r_1 & r_2 & r_3 & r_4 & r_5 \end{array} \\
\left(\begin{array}{ccccc}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0
\end{array}\right)
\end{array}
$$

(b) ILP matrix

Figure 7.7: ILP constraints and corresponding matrix for O_p set shown in Figure 7.6.

$$
\begin{array}{rcll}
r_1 + r_2 & \geq & 1 & (O_2) \\
r_1 & \geq & 1 & (O_4) \\
r_4 + 0.5 \cdot r_2 + 0.5 \cdot r_1 & \geq & 1 & (O_5) \\
r_2 + r_3 + r_5 & \geq & 1 & (O_6) \\
r_2 + r_4 & \geq & 1 & (O_7) \\
r_3 + 0.5 \cdot r_2 + 0.5 \cdot r_1 & \geq & 1 & (O_8)
\end{array}
$$

(a) ILP constraint equations with Partial coverage

$$
A_f = \begin{array}{c} \\ O_2 \\ O_4 \\ O_5 \\ O_6 \\ O_7 \\ O_8 \end{array}
\begin{array}{c}
\begin{array}{ccccc} r_1 & r_2 & r_3 & r_4 & r_5 \end{array} \\
\left(\begin{array}{ccccc}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0.5 & 0.5 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0.5 & 0.5 & 1 & 0 & 0
\end{array} \right)
\end{array}
$$

(b) ILP matrix with Partial Coverage

Figure 7.8: ILP constraints and corresponding matrix for O_p set considering partial coverage.

Chapter 8

Experimental Setup

This Chapter of the thesis details the experimental setup used to test the proposed algorithms/methodologies from Chapter 4, 5, 6, and 7.

8.1 Thermal-Aware Floorplanner

The thermal-aware floorplanner is implemented in C++. It uses a simulated annealing algorithm with a sequence-pair (SP) representation [58]. Thermal analysis is integrated into the floorplanner using Hotpsot 4.1 with the default parameters. The faster block mode is used for optimization and the slower, more accurate grid mode for final results. The experiments are run on a CentOS 5.1 Linux system with a 2.6GHz AMD Opteron processor and 8GB of memory.

The GSRC and MCNC benchmarks are used for experiments. Both the GSRC and MCNC benchmarks do not have actual dimensions or power information and both of these parameters dramatically affect the performance of the heat sink and overall chip cooling. For benchmark dimensions, all the GSRC and MCNC benchmarks are scaled to

be in range of medium to large area chips ($0.5cm^2$ - $2cm^2$). Hence, the dimensions for the MCNC benchmarks are assumed to be in microns and the GSRC benchmarks are in tenths of a micron. The aspect ratio of soft blocks is constrained to the limits specified in the respective benchmarks (0.3 to 3.0). Also, the area is limited so that blocks can not increase their area past 50% of their original area. These constraints are necessary since highly rectangular aspect ratios become increasing difficult to route and blocks that are very large will also increase the wirelengths within the blocks. Also, having too high of a block utilization will require significant buffering to drive signals across the block. In the case of hard blocks, large halos tend to increase wirelengths between blocks.

For block power information, power numbers are randomly generated using power densities similar to the predicted 65nm node in [51]. The power densities used are $750\frac{W}{cm^2}$, $250\frac{W}{cm^2}$ and $25\frac{W}{cm^2}$ with corresponding frequencies of 15%, 45% and 40%. The mean is therefore $235\frac{W}{cm^2}$ and there is approximately a $3.2\times$ difference between the average and maximum power density as observed in [51].

During initial experiments, it was noticed that floorplans with more whitespace tend to have lower temperatures. This is due to two factors: a larger chip will decrease the chance that two hot blocks are close together and a larger chip area will improve the thermal conductivity to the heat sink which results in lower maximum and average temperatures. Consequently, for fair comparisons, fixed-area floorplanning is performed. The area cost during annealing is the area outside of the fixed area. However, there are no constraints on the floorplan aspect ratio. All the experiments used a fixed area that is 10% larger than the total area of all blocks. All the results presented are mean values for 100 simulated annealing runs.

8.2 Floorplanning and Bump Placement Co-Optimization

The floorplanner used for these experiments is similar to the one used for thermal aware floorplanning. The GSRC and MCNC benchmarks are also employed for these experiments and the benchmark dimensions and block power information are similar to the ones used for the thermal-aware floorplanner.

The thermal-stress constants used are as follows: For the Knecht-Fox model C is set to 8.9 as reported in [65]. For the Shear-strain model the constants are set with $C1$ as 501.3, $C2$ as 0.031, $C3$ as 4.96 and $C4$ as 5433.5 which are the material constants reported in [85]. CTE values of 2.3 and 25 are used for the values of $\alpha_c$and α_s respectively.

The positions of the C4 bumps used for the bump placement are constrained to a grid with a minimum bump pitch which of 100μm [92]. The diameter of the solder bumps were set to 50μm and the height of solder bumps were set to 60μm, The thermal conductivity and electrical resistivity of the bumps were assumed to be that of Tin (Sn) and set to 67 W/mK and $1.09 \times 10^{-7}\Omega \cdot m$, respectively. The number of C4 bump locations within the grid is actually larger than the number of I/O bumps required by the chip so as to allow for some flexibility in their placement. The bump temperatures were obtained accurately through Hotspot [82] multi-layer simulations. Three layers are used for the bump simulations: Layer 1 is the bump layer, layer 2 is the silicon layer and the final layer is the thermal interface material (TIM) layer. The current in the bumps are assumed to be between 0.1A and 1A [9].

8.3 Decap Redistribution

The decap redistribution algorithm is implemented in C++. HotSpot 5.0 [82] is used for thermal analysis. The direct solver CHOLMOD (Cholesky factorization) from the UFsparse matrix packages [19] is used for transient power grid analysis. The transient solver is implemented using the Backward Euler Method with a time step of 5ps. The results are obtained on a Ubuntu 10.04 Linux system with a 3.4GHz Intel i7-2600 processor and 8GB of memory.

The IBM power grid transient benchmarks [35] are used for the experiments. These benchmarks are transient extensions of the widely used DC IBM power grid benchmarks [60]. Since no dimensions are given in the benchmarks, they are scaled so that each chip has an average power density of 250W/cm^2 [51]. The temperature map for each benchmark is computed by partitioning each benchmark into blocks. The blocks are extracted from the benchmark based on current source values. Distinct regions within the benchmark have similar current draws and these regions are assumed to be blocks. The total current for each block is summed from the current sources within the block and the power is calculated assuming $V_{dd} = 1.8V$ for all the benchmarks. The RMS current obtained from the current pulse information is used to calculate the power value for each block. The block coordinates, dimensions and powers are then fed to HotSpot for thermal simulation. The value of δ_{add} for the decap redistribution algorithm is set to 1%.

The parameters from a 45nm technology were used to calculate the temperature of the wires [36]. The width of the intermediate wires and global wires are set to 70nm and 100nm, respectively. K_{eff} is set to an average of 5Wm^{-1}K^{-1}. The inter-layer dielectric

thickness is set to 110nm for the intermediate layers and 215nm for the global layers. The thermal spreading factor (ϕ) is set to 0.88 [6]. The activation energy (Q) value used for MTTF comparisons is set to 0.5eV [6].

8.4 Redundant Bump Placement

The redundant bump placement algorithm is implemented in C++. Hotspot 5.0 [82] is used for thermal analysis with the default parameters. The grid mode is used for more final thermal simulations. The direct solver CHOLMOD (Cholesky factorization) from the UFsparse matrix packages [19] is used for final power grid analysis, while an iterative solver is used to generate the coverage sets and also the V_p, F_p and C_p sets. The iterative solver is the preconditioned conjugate gradient method using an incomplete LU factorization as the preconditioner. The initial solution for the solver is the solution with all bumps in place, and the incomplete LU factorization is generated from the conductance matrix with all bumps in place. The iterative solver is used to speed up the power grid simulations required to generate the various sets while maintaining a tolerable residual error. GPLK [25] is used for solving the ILP. The results are obtained on a Ubuntu 10.04 Linux system with a 3.4GHz Intel i7-2600 processor and 8GB of memory.

The IBM power grid benchmarks [60] are used for our experiments. Since no dimensions are given in the benchmarks, we scaled them so that each chip would have an average power density of 250 W/cm^2 [51]. The temperature map for each benchmark was computed by partitioning each benchmark into blocks. The total current for each block is summed from the current sources within the block and the power is calculated assuming

$V_{dd} = 1.8V$ for all the benchmarks. The block coordinates, dimensions and powers are then fed to HotSpot for thermal simulation. The IBM power grid benchmarks also do not contain any capacitance information which is required for transient simulations. Consequently, decoupling capacitance was added to each benchmark to eliminate all transient violations using a conjugate gradient method based on sensitivity analysis [84].

For the electromigration calculations the following values from [14] are used: Q = 0.8eV, n = 1.8 and A = 2.54×10^{-8}. The C4 bump diameter is set to 50 microns.

Chapter 9

Experiments

This chapter of the thesis details the experiments and results for the the proposed thermal-aware CAD algorithms/methodologies in Chapter 4, 5, 6, and 7.

9.1 Thermal-Floorplanning Experiments

This section details the thermal-aware floorplanning experiments corresponding to the floorplanner introduced in Chapter 4. First, a comparison with respect to speed and solution quality between a typical thermal-aware floorplanner and the proposed floorplanner is presented. Then the effectiveness of whitespace allocation on reducing temperatures is reported.

9.1.1 Fast Thermal Floorplanning

One advantage of the proposed thermal-aware floorplanner is the decrease in execution time as a result of evaluating a power-density cost vs a temperature cost (Section 4.2.1.3). Consequently the first experiment for the proposed thermal floorplanner,

compares the execution times of using a power spreading cost and a typical max temperature metric as used in most previous floorplanners [7, 17, 33, 49, 62]. The results of these experiments are depicted in Table 9.1 and clearly show the proposed power metric is effective for thermal-aware floorplanning in terms of reduced execution times. Columns 1, 2 and 3 show the results of doing only HPWL optimization which is used as a baseline for comparison purposes. Columns 4, 5, and 6 show the results of HPWL optimization along with the proposed power density cost metric, and finally columns 7, 8 and 9 show the results of doing HPWL optimization along with a maximum temperature metric like prior works [17, 26, 27, 33, 34, 91, 114].

For the larger benchmarks, the integrated thermal simulation is too slow to finish in a reasonable amount of time and are thus not reported. In contrast, the proposed power-density is very fast even when compared to HPWL-only optimization. The maximum temperatures, however, are comparable or better than the direct temperature optimization results in all cases. The improved results are due to the global view of the proposed cost function when compared to the other, more direct temperature optimization method. The proposed method may accept a move that improves the temperature in a region that is not the highest temperature, but soon becomes a hotspot whereas the temperature-direct method would reject the move. This allows the proposed optimization to get more useful work out of the random SA moves and achieve improved results.

9.1.2 Dynamic Whitespace Allocation

The second experiment investigates the effectiveness of whitespace allocation (dynamic) during floorplanning for hard and soft blocks as detailed in Section 4.2.2.2. The

Table 9.1: Peak Temperature Optimization Results Showing That The Proposed Power Metric Can Significantly Reduce Floorplan Temperatures

	HPWL Only			HPWL and Power			HPWL and Temp		
Benchmark	HPWL	Max Temp (K)	Time (s)	HPWL	Max Temp (K)	Time (s)	HPWL	Max Temp (K)	Time (s)
n10	43653	448.40	2.87	44598	430.40	2.97	44378	440.67	108.35
n30	131170	401.33	8.40	132393	389.86	9.02	134340	389.10	1724.50
n50	172521	414.84	15.16	175237	404.88	15.59	179270	406.35	7082.00
n100	286352	394.63	28.62	290480	389.41	28.54	n/a	n/a	n/a
n200	545746	396.92	88.11	559340	393.07	114.50	n/a	n/a	n/a
n300	799656	397.83	191.05	815146	395.54	266.92	n/a	n/a	n/a
ami33	82287	358.54	10.68	84206	357.87	11.37	84627	358.26	2212.10
ami49	1190385	455.47	22.00	1192307	449.99	23.65	1262700	453.29	6495.20
apte	641408	453.27	4.89	648447	444.89	5.16	656604	446.95	92.73
hp	202489	444.36	5.23	213080	432.86	5.27	209961	439.17	133.90
xerox	540676	366.75	11.64	540177	362.03	11.91	546782	361.97	119.13
Mean Change	0	0.00	0.00	**1.8%**	**7.41**	**1.12**×	**3.0%**	**5.90**	**158.44**×

results of these experiments are summarized in Table 9.2 which shows that an almost 2× peak temperature decrease can be obtained by adjusting the whitespace utilization around hard blocks when compared to just using a power density metric. For soft blocks, the results are not as impressive with only a slight improvement in peak temperatures when compared to just using the power density metric.

Table 9.2: Dynamic Whitespace Utilization Peak Temperature Results for Hard and Soft Blocks Showing Further Reduction in Floorplan Temperatures

Experiment	HPWL	Max Temp (K)	Time (s)
HPWL and Power	1.80%	7.41	1.11×
HPWL and Temp	3.00%	5.9	158.44×
HPWL, Power and Area Utilization (Soft)	4.20%	**7.60**	1.12×
HPWL, Power and Area Utilization (Hard)	3.30%	**9.16**	1.14×

9.1.3 Static Whitespace Allocation

The third experiment investigates the effectiveness of whitespace allocation (static) during floorplanning for hard and soft blocks as detailed in Section 4.2.2.1. The available whitespace for each benchmark is 10% of the total area due to the fixed area requirement discusses in Section 8.1. Hence, 50% of the whitespace available (corresponding to 5% of the total area) is used as available whitespace. Using 100% of the whitespace available would require the floorplans to be perfect (i.e., contain no additional white space), whereas using 0% of the available whitespace would lead to no improvements.

The results of the static whitespace allocation experiments are detailed in Table 9.3. The results show that static allocation does better than dynamic area allocation on

average for all benchmarks. This occurs because increasing the area of a block will always decrease its maximum temperature. Consequently, dynamic allocation of the whitespace during floorplanning results in low power density blocks being inflated even though the whitespace might be best use for a high power density block. In addition, the usage of the SA random moves to adjust the whitespace usage detracts from more useful moves that reduce area, HPWL, and minimize the adjacency of high power blocks. The results for hard blocks are not shown since the inflated size of the hard blocks resulted in most floorplans not meeting the fixed area requirement.

Table 9.3: Static Whitespace Utilization Peak Temperature Results for Soft Blocks Showing Increased Reduction in Floorplan Temperatures

Experiment	HPWL	Max Temp (K)	Time (s)
HPWL and Power	0.20%	5.94	1.14×
HPWL and Temp	1.80%	6.12	158.4×
HPWL, Power and Area Utilization	6.40%	**11.44**	1.12×

9.1.4 Example Floorplanning Result

An example plot of n100 with and without temperature optimization is shown in Figure 9.1 with identical fixed-areas. Figure 9.1(a) corresponds to HPWL-only optimization. The HPWL for this placement is 278431, the maximum temperature is 400.1K and it required 24.6s computing time. Figure 9.1(b) corresponds to HPWL, power spreading and static whitespace optimization. Its HPWL is 310124, the maximum temperature is 382.7K and required 28.6s computing time. These figures show that utilizing area allocation along with the proposed power density metric, significantly reduces peak temperature with

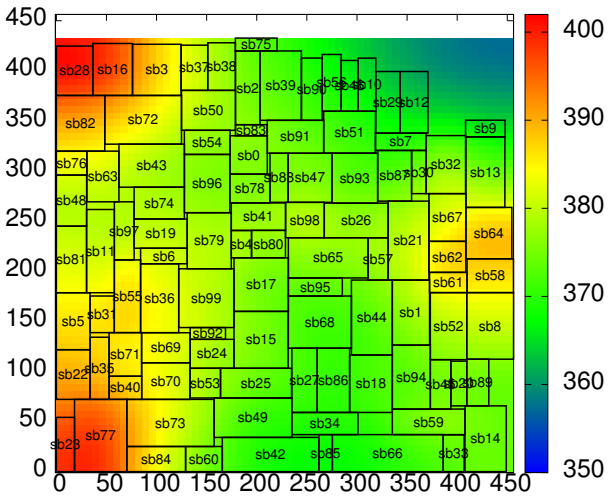
a modest increase in HPWL and a very small increase in run-time. The placement without whitespace utilization tends to cluster unused area in the upper-right of the floorplan due to the sequence pair representation. The placement that has whitespace utilization, distributes available whitespace to the internal floorplan blocks which results in fewer hotspots being created.

9.2 Floorplanning and Bump Placement Co-optimization

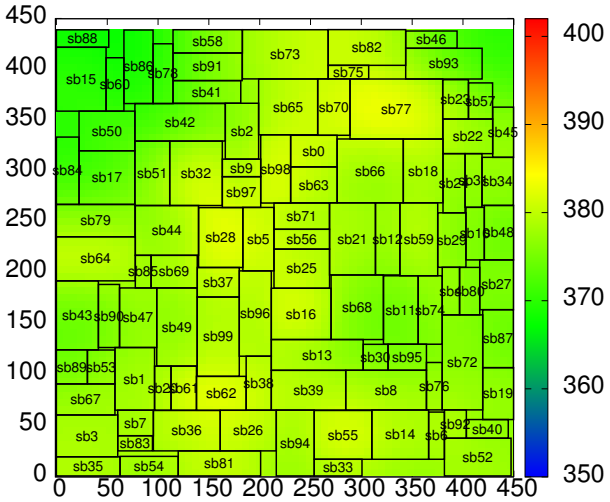
This section details the experiments for increasing the lifetime of C4 bumps corresponding to the floorplanning and bump placement co-optimization methodology presented in Chapter 5. Three sets of experiments are used to analyze the effectiveness of the proposed floorplanning and bump placement co-optimization methodology. The first set of experiments used as a baseline, only does HPWL optimization. The second sets of experiments does only thermal-aware floorplanning, and the last set of experiments does thermal floorplanning and bump placement co-optimization.

9.3 Baseline Experiment - HPWL Optimization

The first experiment consisted of only doing HPWL optimization (no pruning of candidate bump locations) to serve as a baseline for the other two sets of experiments. The results of the HPWL optimization are shown in Table 9.4 in columns 1-5. The creep rate ($\dot{\epsilon_c}$) reported in the table is the maximum for all the bumps, and the number of cycles to failure (N_f) is calculated from that creep rate. All benchmarks have a number of cycles to failure that are less than 400.



(a) HPWL Optimization Only. Circuit dimensions in μm and temperature in Kelvin (K)



(b) HPWL, Power Spreading and Static Whitespace Optimization. Circuit dimensions in μm and temperature in Kelvin (K)

Figure 9.1: Example results for n100 benchmark showing how significant reductions in floorplan temperatures by using the proposed thermal-aware floorplanning methods.

9.3.1 Temperature Optimization

The second set of experiments consisted of doing HPWL optimization in conjunction with temperature optimization (no pruning). The results of these experiments are shown in Table 9.4 in columns 6-10. Temperature optimization is able to decrease the average maximum creep rate over all benchmarks by 85% and increase the lifetime of C4 balls by a factor of $12\times$ even with just a 3K decrease in peak temperature. This significant decrease can be explained by examining the location of hotspots in HPWL vs. Temperature optimized floorplans. In HPWL optimized floorplans, blocks with large power densities have a possibility of being placed on the periphery of the chip causing significant hotspots due to the adiabatic boundary conditions. Consequently, if a bump is placed in the vicinity of that block it will have a high creep rate, since it is located at the edge of the chip and is subjected to high temperatures. In temperature optimised floorplans, blocks with large power densities have a much lower probability of being placed at the periphery of the chip since that would lead to large temperatures. Consequently, there are fewer hotspots located on the periphery of the chip, the area of the chip that is susceptible to large creep rates.

9.3.2 Thermal Floorplanning with Bump Placement

The final set of experiments consists of doing HPWL and temperature optimization concurrently with the quadratic bump placer (with pruning). The results of these experiments in Table 9.5 show that co-optimization can lead to significant improvements in the C4 bump reliability, even when compared to thermal-aware floorplanning. The co-optimization decreased the average maximum creep rate over all benchmarks by a factor of 94% and increase the lifetime of C4 balls by a factor of $47\times$ as compared to HPWL only op-

Table 9.4: Temperature Optimization Results Showing Increase in Bump Reliability

	HPWL Only					HPWL and Temperature Optimization				
Bench.	HPWL	Max T (K)	$\dot{\epsilon_c}$ (1/s)	N_f	Time (s)	HPWL	Max T (K)	$\dot{\epsilon_c}$ (1/s)	N_f	Time (s)
n30	54420	364.64	1.44E-02	609	9.39	54899	360.29	5.37E-04	16367	9.69
n50	103339	368.68	2.62E-02	336	14.57	104039	365.42	1.44E-03	6100	16.67
n100	177711	357.87	3.17E-03	2770	27.46	179513	355.37	5.83E-04	15077	32.56
n200	371960	357.68	2.65E-03	3312	95.96	374971	356.21	7.36E-04	11946	128.15
n300	596528	359.49	4.28E-03	2054	185.94	603540	358.34	8.88E-04	9903	283.84
Mean	0%	0K	0 1/s	0×	0×	1.7%	2.55K	**0.15**×	**12**×	1.25×

timization. These improvements came without a significant increase in HPWL wirelength (3.6%) or runtime (1.36×) as compared to HPWL only optimization.

Table 9.5: Bump Placement Optimization Results Showing Added Increase in Bump Reliability

	HPWL, Temperature and Reliability Optimization				
Bench.	HPWL	Max T	$\dot{\epsilon_c}$	N_f	Time (s)
n30	59230	357.49	1.06E-04	82925	10.58
n50	107321	364.29	3.89E-04	22597	17.35
n100	182905	353.59	1.94e-04	45310	33.98
n200	379144	355.14	3.26e-04	26964	142.63
n300	600652	358.38	4.90e-04	17939	327.58
Mean	3.6%	3.96 K	**0.06**×	**47**×	1.36×

9.3.3 Example Floorplanning and Bump placement Result

An example floorplan of n100 benchmark with and without floorplanning and bump placement co-optimization (reliability floorplanning) is shown in Figure 9.2. Figure 9.2(a) corresponds to HPWL-only optimization. The HPWL for this placement is 176537, the maximum temperature is 358.5K, the maximum creep rate is 1.64E-3, and the runtime is 26.16s. The creep rate corresponds to a number of cycles to failure value of 5349. Figure 9.2(b) corresponds to reliability floorplanning. The HPWL for this placement is 179834, the maximum temperature is 352.9K, the maximum creep rate is 1.40E-4 and the runtime is 29.86s. The creep rate corresponds to a number of cycles to failure of 63083. These figures show that by using reliability floorplanning, the lifetime of C4 balls can be

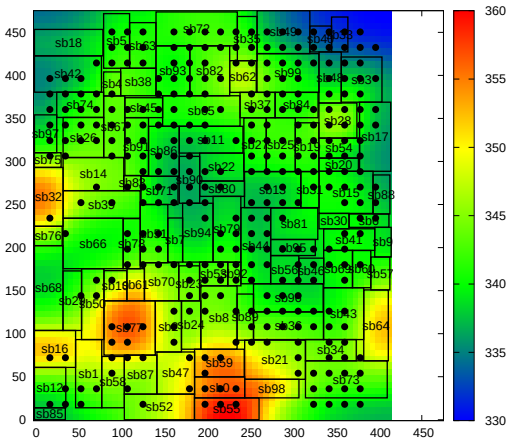
greatly improved with a modest increase in HPWL and run-time. The reliability floorplanning tends to not place bumps in hotspots, especially those located on the edge, while the HPWL optimization will tend to place bumps closer to their respectively blocks even if it means placing a bump in a position where it can fail rapidly due to thermal cycling.

9.4 Decap Redistribution

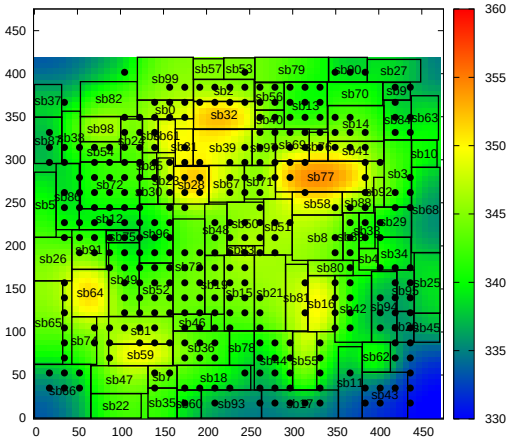
This section details the experiments used to demonstrate the effectiveness of the proposed decap redistribution algorithm in reducing interconnect temperatures. The first experiment (Section 9.4.1) confirms the large wire temperature increases when decap placement doesn't consider Joule heating. The second experiment (Section 9.4.2) evaluates the effectiveness of the proposed Joule heating-aware decap redistribution at reducing interconnect temperatures. The third experiment (Section 9.4.3) evaluates the additional gains by optimizing maximum interconnect temperature directly instead of minimizing Joule heating. The final experiment (Section 9.4.4) quantifies the extent that additional decap beyond that for voltage droop decap can improve thermal reliability of PSN interconnect.

9.4.1 Baseline Experiment

The results of the baseline experiment highlighting the effects of Joule heating on PSN interconnect for the IBM transient benchmarks are shown in Table 9.6. The J_w column represents the sum of all the Joule heating RMS values for the wires within the benchmark. The Max T column represents the largest wire temperature. The Avg ΔT and Max ΔT columns represent the average and maximum change in wire temperatures



(a) HPWL Optimization Only. Circuit dimensions in μm, temperature in Kelvin (K)



(b) HPWL, Temperature and Bump Optimization. Circuit dimensions in μm, temperature in Kelvin (K)

Figure 9.2: Example Results for n100 Showing Increase in Bump Reliability (100×) Using Proposed Methods.

caused by Joule heating respectively. Finally, the ΔR and ΔMTTF columns represent the maximum increase in resistivity and maximum decrease in the electromigration lifetime

Table 9.6: Joule heating reduces the PSN interconnect electromigration lifetime by up to $0.12\times$.

Bench	J_w (W)	Max T (K)	Avg ΔT (K)	Max ΔT (K)	ΔR ($\times$)	ΔMTTF ($\times$)
ibmpg1t	2.26	423.1	6.5	56.4	1.2	**0.12**
ibmpg2t	1.97	374.7	1.1	9.0	1.0	**0.68**

respectively. The large values for ΔR and ΔMTTF especially for the ibmpg1 benchmark clearly shows that Joule heating in PSN interconnect can severely affect the reliability and robustness of the PSN.

It should be noted that while the total Joule heating RMS power is small compared to the total chip power (about 30W for both benchmarks), it is significant since the area for thermal diffusion (cross-sectional area of wires) is small. The large values for Max ΔT across both benchmark demonstrate the significance of the Joule heating power.

9.4.2 Decap Redistribution to Minimize Total Joule Heating Power

The second experiment investigates the effectiveness of the proposed gradient-based Joule heating aware decap redistributing algorithm in reducing interconnect temperatures and subsequent reliability problems in the IBM transient benchmarks. For these experiments, C_{rem} was set to 10% of the total decap within each partition.

The results of these experiments in Table 9.7 show that the proposed algorithm can reduce the total Joule heating and consequently the temperature of PSN interconnect leading to increase reliability from electromigration and decreases resistivity. The ΔR

and ΔMTTF column shows the improvements in resistivity and electromigration lifetime failure rate in the wire with the maximum increase in resistivity and the maximum decrease in electromigration in the baseline experiment with no decap redistributed. The increase in electromigration lifetime failure rate is significant across both benchmarks (1.39× on average).

Table 9.7: Joule heating-aware decap redistribution increases interconnect electromigration reliability by an average of $1.39\times$.

Bench	J_w (W)	Max T (K)	Max ΔT (K)	ΔR (×)	ΔMTTF (×)	Time (sec)
ibmpg1t	1.76	402.2	38.3	0.96	1.74	97
ibmpg2t	1.77	373.2	8.9	0.997	1.04	1549
Improvement	**16.1%**	**11.2K**	**21.9%**	**0.98×**	**1.39×**	

9.4.3 Decap Redistribution to Minimize ΔT

The third experiment investigates the effectiveness of optimizing for ΔT as opposed to Joule heating power. Minimizing Joule heating power does not guarantee minimal increase in interconnect temperatures since temperature also depends on the thermal resistance to the substrate. The results of this experiment in Table 9.8 show that the interconnect temperature increases can be further reduced when considering ΔT leading to increased reliability as measured in electromigration lifetime failure rate. The amount of Joule heating reduced is less than that for the Joule heating aware decap distribution, however the max interconnect temperature, resistivity decrease and electromigration lifetime failure rate in-

crease is larger than that for the Joule heating aware decap distribution. Figure 9.3 shows the final wire temperatures from the two algorithms and highlights the advantages of optimizing for temperature vs. Joule heating. The largest temperature bin for the ΔT optimized decap redistribution is smaller than that for the Joule heat optimized decap redistribution.

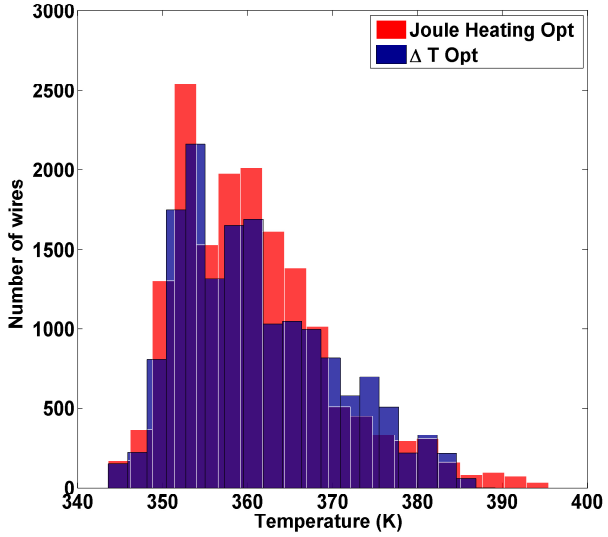


Figure 9.3: The ibmpg1t benchmark illustrates that the ΔT optimized decap redistribution has fewer wires in the high temperature bins as compared to the Joule heat optimized decap redistribution.

Table 9.8: Temperature-aware decap redistribution increases interconnect electromigration reliability by an average of $1.66\times$.

Bench	J_w (W)	Max T (K)	Max ΔT (K)	ΔR ($\times$)	ΔMTTF ($\times$)	Time (sec)
ibmpg1t	1.79	399.8	31.9	0.94	2.24	124
ibmpg2t	1.81	373.0	7.1	0.99	1.08	1679
Improvement	**14.5%**	**12.5K**	**32.2%**	**0.97**$\times$	**1.66**$\times$	

9.4.4 Additional Decap

For the final experiment, C_{budget} is created by adding additional decap the design as opposed to using removed decap from the various partitions For these experiments the additional decap used for C_{budget} is set to 10% of the original total decap within the design. The runtime of the algorithms with the additional decap is less than those in first two experiments. No additional simulations are required to calculate C_{budget} since no decap is removed from any partition and thus no voltage droop violations can occur. Using additional decap greatly reduces total Joule heating RMS power, max interconnect temperature, increases in resistivity and more importantly increases the electromigration mean time to failure. However, adding more decap to a design has several drawbacks and limitations. Decaps are leaky, consequently the additional decaps will increase the total power used by the circuit. More importantly, most modern designs are constrained by space consequently there is not much available area for additional decap. Figure 9.4 demonstrates the decrease in Joule heating for the ibmpg1 transient benchmark caused by the insertion of additional decap.

9.5 Redundant bump placement

This section details the experiments used to demonstrate the effectiveness of the various proposed bump redundancy algorithms. It should be noted that the ibmpg4 benchmark is not used in the experiments due to the small total current in this design. For all experimental results, each power grid is simulated with the redundancy set to ensure there are no electromigration problems or static/dynamic voltage violations.

Table 9.9: Adding 10% more decap increases interconnect electromigration reliability by an average of $2.20\times$.

	Joule Heating Optimization						ΔT Optimization					
Bench	J_w (W)	Max T (K)	Max ΔT (K)	ΔR ($\times$)	ΔMTTF ($\times$)	Time (sec)	J_w (W)	Max T (K)	Max ΔT (K)	ΔR ($\times$)	ΔMTTF ($\times$)	Time (sec)
ibmpg1t	1.52	395.4	28.6	0.93	2.57	51	1.65	389.1	22.9	0.91	3.20	77
ibmpg2t	1.42	371.7	5.9	0.99	1.15	1117	1.44	370.5	4.7	0.99	1.20	1247
Improvement	**30.3%**	**15.4K**	**42.1%**	**0.96**$\times$	**1.86**$\times$		**27.0 %**	**19.1K**	**53.7%**	**0.95**$\times$	**2.20**$\times$	

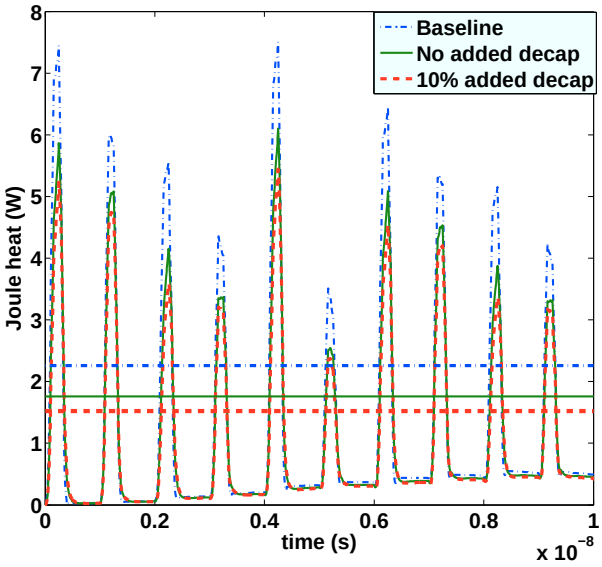


Figure 9.4: An example from the ibmpg1t benchmark shows the decrease in total Joule heating in wires when redistributing decap and adding additional decap. Horizontal lines represent RMS values.

9.5.1 Baseline Experiments

The first set of experiments demonstrate the necessity for single-bump redundancy by showing the voltage and electromigration violations that occur when no redundant bumps are added. The results of this experiment are shown in Table 9.10 under the no redundancy column. The Volt. Viols. column represents percentage of bumps that cause voltage violations if they have a defect/failure and the EM Viols. column represents the percentage of bumps that have an electro-migration violation for any bump defect/failure. The first two columns in Table 9.10 list the total number of candidate bumps and actual bumps in the design respectively. The Ad. Bumps column in the table represent the total number of added bumps for each algorithm.

Table 9.10: Comparison of Different Bump Redundancy Schemes Showing Reduced Redundant Sets Generated by Using the ILP Formulation vs. The Naive Method

			No Redundancy		Naive Greedy (Baseline)		Improved Greedy		ILP	
Benchmark	# Cd. Bumps	# Bumps	EM Viols	Volt. Viols	Ad. Bumps	Time	Ad. Bumps	Time	Ad. Bumps	Time
ibmpg1	6085	269	53.2%	25.7%	198	1.55	46.2%	8.9	44.8%	8.9
ibmpg2	2095	220	65.0%	11.4%	154	18.49	21.4%	133.9	18.6%	134.0
ibmpg3	4423	183	9.8%	32.2%	71	54.91	43.8%	310.0	41.2%	310.0
ibmpg6	22400	84	58.3%	42.9%	64	140.9	38.1%	947.3	38.1%	948.2
mean			46.6%	28.1%			37.4%	6.3x	35.7%	6.4x

9.5.2 Naive Greedy Method

The second set of experiments demonstrate the effectiveness of using the Naive Greedy method for providing single-bump redundancy coverage. The results for these experiments are shown in Table 9.10 under the Naive Greedy column and shows the large number of redundant bumps used for this method, 64.6% on average.

9.5.3 Improved Greedy Method and ILP

The third set of experiments compares and contrasts the two novel methods for generating redundant bump sets, the improved greedy and ILP method. The size of the R_p sets generated for these method are much smaller than those using the Naive Greedy method as shown in Table 9.10, however, the runtime increases significantly due to the time to taken to calculate the coverage sets. The Improved Greedy Method tends to have a slightly shorter runtime compared to the ILP method but is not always optimal. The ILP method, which is optimal, universally creates the smallest R_p sets.

9.5.4 Partial Coverage

The final sets of experiments extend the proposed ILP algorithm to consider partial coverage sets as detailed in Section 7.3.3. The size of the R_p sets considering partial coverage is smaller that those which do not consider partial coverage, but the runtime is significantly increased. The runtime increases are due to the denser ILP matrix which leads to much longer solving times for the ILP solver. The runtime for the largest benchmark ibmpg6 is less than 30 minutes which is acceptable since the PSN for this benchmark contains more than one million nodes.

Table 9.11: Bump Redundancy Set Generation using Partial Coverage Showing Decrease in Size of Redundant Sets When Partial Coverage is Considered

Benchmark	Added Bumps %	Runtime
ibmpg1	44.8%	42.83
ibmpg2	15.7%	238.22
ibmpg3	36.1%	311.81
ibmpg6	38.1%	1261.62
mean	33.7%	13.8x

Chapter 10

Conclusion

High on chip temperatures have recently become a major concern for IC designers. These large temperatures significantly decrease the reliability of ICs, increase power consumption and increase packaging costs. This thesis investigates several thermal-aware CAD methodologies for addressing specific reliability issues caused by high on chip temperatures.

10.1 Thesis Contributions

First, a thermal-aware floorplanning methodology was proposed that addresses two of the main deficiencies of previous thermal-aware floorplanners found in literature: long runtimes and inadequate floorplanning moves for reducing temperature. The methodology uses a power density metric as a means of guiding floorplanning as oppose to doing direct thermal simulations which significantly decreases the runtime required for floorplanning. In addition, the methodology introduces new floorplanning moves based on whitespace utilization to reduce high on chip temperatures. The new methodology is able to reduce

on chip temperatures for the GSRC benchmarks by 7K with only a 4.2% increase in HPWL and $1.14\times$ increase in runtime.

Second, a package-chip co-design thermal-aware floorplanning methodology was proposed to consider the effects that high on-chip temperatures have on C4 bump reliability. In addition, a thermal fatigue model is proposed that can be used for chip package co-optimization to quickly evaluate the thermal fatigue of package bumps. The model is used to quickly evaluate candidate C4 bump locations to guide reliability floorplanning. The reliability floorplanner is able to significantly increase the lifetime of C4 bumps, even when compared to thermal-aware floorplanning. Thermal-aware floorplanning was able to increase the lifetime of C4 bumps by $12\times$ on average compared to only HPWL optimization. However, the proposed quadratic pin placement algorithm was able to improve on thermal-aware floorplanning significantly, as the increase in the lifetime of C4 bumps was $49\times$ on average compared to only HPWL optimization. These improvements came with a modest 3% increase in wirelength and a $1.36\times$ increase in runtime.

Third, a methodology of reducing Joule heating in PSN interconnect by redistributing decap using a gradient based method was proposed. Experiments show that the algorithm is able to reduce interconnect temperatures on average by 12.5K which results in a decrease in resistivity by a factor of $0.97\times$ and an increase of electromigration lifetime by a factor of $1.66\times$. The methodology is extended to consider placing 10% additional decap which results in reduced interconnect temperatures of 19.1K corresponding to a a decrease in resistivity by a factor of $0.96\times$ and increase in electromigration lifetime by a factor of $2.20\times$.

Finally, a methodology for combating the pin electromigration problem exacer-

bated by high on chip temperatures was proposed. The methodology, uses and ILP formulation for generating a redundant power pin set to guarantee single-pin redundancy. The ILP formulation is able to generate redundant pin sets for the IBM power grid benchmarks using 68% fewer additional pins than a Naive greedy method on average.

10.2 Future Work

This thesis proposes several thermal-aware methodologies for specific reliability issues cause by high on chip temperatures. However, there is still significant research to be done within the area of thermal-aware VLSI CAD. The single-bump redundancy placement methodology can be extended to consider multiple bump failures. In addition, the decap placement methodology to reduce joule heating can be extended to consider power gating. In addition to these extensions, another area of additional research is creating algorithms for reducing NBTI. It should be noted that there has been significant research in reducing NBTI using different chemical components in PMOS devices. Another area of interest is the effect that high temperatures and large temperature gradients can have on critical circuit elements such as the clock distribution network. Finally, another interesting area of thermal-aware CAD research is reducing leakage currents in circuits due to the constant shrinking of power budgets.

Bibliography

[1] S.N. Adya and I.L. Markov. Fixed-outline floorplanning through better local search. In *Proceedings of 2001 International Conference on Computer Design*, pages 328 –334, 2001.

[2] S.N. Adya and I.L. Markov. Consistent placement of macro-blocks using floorplanning and standard-cell placement. In *Proceedings of the 2002 International Symposium on Physical design*, pages 12–17, 2002.

[3] S.N. Adya and I.L. Markov. Fixed-outline floorplanning: enabling hierarchical design. *IEEE Transactions on VLSI Systems*, 11(6):1120–1135, December 2003.

[4] M.A. Alam and S. Mahapatra. A comprehensive model of PMOS NBTI degradation. *Microelectronics Reliability*, 45(1):71 – 81, 2005.

[5] A. Alvandpour, P. Larsson-Edefors, and C. Svensson. Separation and extraction of short-circuit power consumption in digital CMOS VLSI circuits. In *Proceedings of 1998 International Symposium on Low Power Electronics and Design*, pages 245 –249, August 1998.

[6] K. Banerjee and A. Mehrotra. Global (interconnect) warming. *IEEE Circuits and Devices Magazine*, 17(5):16–32, September 2001.

[7] Y. Cai, B. Liu, Q. Zhou, and X. Hong. A thermal aware floorplanning algorithm supporting voltage islands for low power SOC design. In *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation, Lecture Notes in Computer Science*, volume 3728, pages 909–909. 2005.

[8] Y.C. Chang, Y.W. Chang, G.M. Wu, and S.W. Wu. B*-trees: a new representation for non-slicing floorplans. In *Proceedings of the 37th Conference on Design Automation*, pages 458–463. ACM, 2000.

[9] D.S. Chau, C. Chiu, J. Torresola, S. Prstic, and S. Reynolds. Experimental method of measuring C4 die bump temperature for electronics packaging. In *Proceedings of The Ninth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 91–95 Vol.1, June 2004.

[10] H.M. Chen, L.D. Huang, I-Min Liu, and M.D.F. Wong. Simultaneous power supply planning and noise avoidance in floorplan design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(4):578 – 587, April 2005.

[11] Y.-K. Cheng et al. ILLIADS-T: An electrothermal timing simulator for temperature-sensitive reliability diagnosis of CMOS VLSI chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17:668–681, 1998.

[12] T.Y. Chiang, B. Shieh, and K.C. Saraswat. Impact of joule heating on scaling of

deep sub-micron Cu/low-k interconnects. In *Symposium on VLSI Technology*, pages 38–39, 2002.

[13] E. Chiprout. Fast flip-chip power grid analysis via locality and grid shells. In *Proceedings of the 2004 IEEE/ACM international conference on Computer-aided design*, pages 485 – 488, 2004.

[14] W.J. Choi, E.C.C. Yeh, and K.N. Tu. Mean-time-to-failure study of flip chip solder joints on Cu/Ni(V)/Al thin-film under-bump-metallization. *Journal of Applied Physics*, 94:5665–5671, 2003.

[15] C.-T. Chu et al. Temperature aware microprocessor floorplanning considering application dependent power load. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 586–589, 2007.

[16] J. Cong, M. Romesis, and J. R. Shinnerl. Fast floorplanning by look-ahead enabled recursive bipartitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1719–1732, 2006.

[17] J. Cong, J. Wei, and Y. Zhang. A thermal-driven floorplanning algorithm for 3D ICs. In *Proceedings of the 2004 IEEE/ACM international conference on Computer-aided design*, pages 306–313, 2004.

[18] D. Cuesta, J. L. Risco-Martin, J. L. Ayala, and J. I. Hidalgo. 3D thermal-aware floorplanner using a MOEA approximation. *Integration, The VLSI Journal*, 46(1):10–21, 2012.

[19] T.A. Davis. Ufsparse. `http://www.cise.ufl.edu/research/sparse/`.

[20] K. S. Dieter. Negative bias temperature instability: What do we understand? *Microelectronics Reliability*, 47(6):841 – 852, 2007.

[21] J.W. Fang, I.J. Lin, Y.W. Chang, and J.H. Wang. A Network-Flow-Based RDL Routing Algorithmz for Flip-Chip Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(8):1417–1429, August 2007.

[22] D. R. Frear, J. W. Jang, J. K. Lin, and C. Zhang. Pb-free solders for flip-chip interconnects. *JOM Journal of the Minerals, Metals and Materials Society*, 53(6):28 – 33, 2001.

[23] J. Fu, Z. Luo, X. Hong, Y. Cai, Z. Pan, and S.X.-D. Tan. A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery. In *Proceedings of the 2004 Asia and South Pacific Design Automation Conference*, pages 505–510, 2004.

[24] J. Fu, Z. Luo, X. Hong, Y. Cai, Z. Pan, and S.X.-D. Tan. VLSI on-chip power/ground network optimization considering decap leakage currents. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 735 – 738, January 2005.

[25] GLPK. `http://www.gnu.org/software/glpk/`.

[26] B. Goplen and S. Sapatnekar. Efficient thermal placement of standard cells in 3D ICs using a force directed approach. In *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 86, 2003.

[27] A. Gupta, N.D. Dutt, F.J. Kurdahi, K.S. Khouri, and M.S. Abadir. LEAF: A system

level leakage-aware floorplanner for SoCs. In *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, pages 274–279, January 2007.

[28] S.P. Gurrum, S.K. Suman, Y.K. Joshi, and A.G. Fedorov. Thermal issues in next-generation integrated circuits. *IEEE Transactions on Device and Materials Reliability*, 4(4):709–714, December 2004.

[29] D. Harris and N.H.E. Weste. *CMOS VLSI Design*. Addison-Wesley, 2005.

[30] V. M. Heriz et al. Method of images for the fast calculation of temperature distributions in packaged VLSI chips. *13th International Workshop on Thermal Investigation of ICs and Systems*, pages 18–25, 2007.

[31] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, and C.K. Cheng. Corner block list: an effective and efficient topological representation of non-slicing floorplan. In *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*, pages 8–12, 2000.

[32] H.Y. Hsieh and T.C. Wang. Simple yet effective algorithms for block and I/O buffer placement in flip-chip design. In *2005 IEEE International Symposium on Circuits and Systems*, pages 1879–1882, May 2005.

[33] W-L. Hung et al. Thermal-aware floorplanning using genetic algorithms. In *Proceedings of 2005 IEEE Symposium on Quality of Electronic Design*, pages 634–639, 2005.

[34] W.-L. Hung et al. Interconnect and thermal-aware floorplanning for 3D micropro-

cessors. In *Proceedings of 2006 IEEE Symposium on Quality of Electronic Design*, pages 98–104, 2006.

[35] IBM. Transient power grid benchmarks. `http://dropzone.tamu.edu/~pli/PGBench/`.

[36] S. Im, N. Srivastava, K. Banerjee, and K.E. Goodson. Scaling analysis of multilevel interconnect temperatures for high-performance ICs. *IEEE Transactions on Electron Devices*, 52(12):2710–2719, December 2005.

[37] S. Jairam and S.K. Roy. Incremental optimization of power pads based on adjoint network sensitivity. In *Proceedings of 2009 Asia Symposium on Quality of Electronic Design*, pages 259 –263, July 2009.

[38] L. Jiang, Y. Cheng, and J. Mao. Analysis and optimization of thermal-driven global interconnects in nanometer design. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 1(10):1564 –1572, October 2011.

[39] A. B. Kahng, B. Liu, and S. X.-D. Tan. Efficient decoupling capacitor planning via convex programming methods. In *Proceedings of the 2006 international symposium on Physical design*, pages 102–107, 2006.

[40] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. In *Proceedings of the 2004 international symposium on Physical design*, pages 18–25, 2004.

[41] C. Kim and D.F. Baldwin. No-flow underfill process modeling and analysis for low

cost, high throughput flip chip assembly. *Transactions on Electronics Packaging Manufacturing*, 26(2):156 – 165, April 2003.

[42] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[43] M. Lai and D. Wong. Slicing tree is a complete floorplan representation. In *Proceedings of the conference on Design, Automation and Test in Europe*, pages 228–232, 2001.

[44] J. H. Lau and S. H. Pan. Creep behaviors of flip chip on board with 96.5Sn-3.5Ag and 100In lead-free solder joints. *The International Journal of Microcircuits and Electronic Packaging*, 24(1):866–873, 2001.

[45] J. Lee. General thermal force model with experimental studies. *Transactions on Packaging*, 29(1):20–29, 2006.

[46] R.-J. Lee, M.-F. Lai, and H.-M. Chen. Fast flip-chip pin-out designation respin by pin-block design and floorplanning for package-board codesign. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pages 804–809, January 2007.

[47] H. Li, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Partitioning-based approach to fast on-chip decap budgeting and minimization. In *Proceedings of the 2005 Design Automation Conference*, pages 170–175, 2005.

[48] X. Li et al. Thermal-aware incremental floorplanning for 3D ICs. *IEEE 7th International Conference on ASIC*, pages 1092–1095, 2007.

[49] Z. Li, X. Hong, Q. Zhou, J. Bian, H. H. Yang, and V. Pitchumani. Efficient thermal-oriented 3D floorplanning and thermal via planning for two-stacked-die integration. *ACM Transactions on Design Automation of Electronic Systems*, 11(2):325–345, April 2006.

[50] J.-M. Lin and Y.-W. Chang. TCG: a transitive closure graph-based representation for non-slicing floorplans. In *Proceedings of the 38th Design Automation Conference*, pages 764–769, 2001.

[51] G. M. Link and N. Vijaykrishnan. Thermal trends in emerging technologies. *Proceedings of the 2006 IEEE Symposium on Quality of Electronic Design*, pages 625–632, 2006.

[52] B. Liu and S.X.-D. Tan. Minimum decoupling capacitor insertion in VLSI power/ground supply networks by semidefinite and linear programs. *IEEE Transactions on Very Large Scale Integration Systems,*, 15(11):1284 –1287, November 2007.

[53] S. Logan and M.R. Guthaus. Fast thermal-aware floorplanning using white-space optimization. In *2009 17th IFIP International Conference on Very Large Scale Integration*, pages 65 –70, October 2009.

[54] S. Logan and M.R. Guthaus. Package-chip co-design to increase flip-chip C4 reliability. In *2011 12th International Symposium on Quality Electronic Design*, pages 1 –6, March 2011.

[55] C.H. Lu, H.-M. Chen, and C.-N. J. Liu. Effective decap insertion in area-array SOC

floorplan design. *ACM Transactions on Design Automation of Electronic Systems*, pages 1–20, 2008.

[56] Y. Ma, S. Dong, X. Hong, Y. Cai, C.K. Cheng, and J. Gu. VLSI floorplanning with boundary constraints based on corner block list. In *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, pages 509–514, 2001.

[57] R.N. Master, A. Marathe, V. Pham, and D. Morken. Electromigration of C4 bumps in ceramic and organic flip-chip packages. In *Electronic Components and Technology Conference*, 2006.

[58] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1518–1524, December 1996.

[59] F.N. Najm. A survey of power estimation techniques in VLSI circuits. *IEEE Transactions on Very Large Scale Integration Systems*, 2(4):446–455, December 1994.

[60] S. R. Nassif. Power grid analysis benchmarks. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, pages 376–381, 2008.

[61] M. Ning, J. Fan, S.X-D. Tan, Y. Cai, and X. Hong. Statistical analysis of on-chip power delivery networks considering lognormal leakage current variations with spatial correlation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(7):2064 –2075, August 2008.

[62] V. Nookala, D. J. Lilja, and S. S. Sapatnekar. Temperature-aware floorplanning of

microarchitecture blocks with IPC-power dependence modeling and transient analysis. In *Proceedings of the 2006 international symposium on Low power electronics and design*, pages 298–303, 2006.

[63] NVIDIA. Second quarter release. `http://www.nvidia.com/object/io\_1215037160521.html`.

[64] J. Oh and M. Pedram. Multi-pad power/ground network design for uniform distribution of ground bounce. In *Proceedings of the 35th annual Design Automation Conference*, pages 287–290, 1998.

[65] J.H.L. Pang and D.Y.R. Chong. Flip chip on board solder joint reliability analysis using 2-D and 3-D FEA models. *IEEE Transactions on Advanced Packaging*, 24(4):499–506, November 2001.

[66] J.H.L. Pang, D.Y.R. Chong, and T.H. Low. Thermal cycling analysis of flip-chip solder joint reliability. *IEEE Transactions on Components and Packaging Technologies*, 24(4):705–712, 2001.

[67] J.-H. Park et al. Fast computation of temperature profiles of VLSI ICs with high spatial resolution. In *Proceedings of the 2008 Semiconductor Thermal Measurement and Management Symposium*, pages 50–54, 2008.

[68] G. Pascariu, P. Cronin, and D. Crowley. Next generation electronics packaging utilizing flip chip technology. In *Proceedings of the 28th International Electronics Manufacturing Technology Symposium*, pages 423–426, July 2003.

[69] M. Pedram and S. Nazarian. Thermal modeling, analysis, and management in VLSI

circuits: Principles and methods. *Proceedings of the IEEE*, 94(8):1487–1501, August 2006.

[70] C.-Y. Peng, W.-C. Chao, Y.-W. Chang, and J.-H. Wang. Simultaneous block and I/O buffer floorplanning for flip-chip design. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pages 213–218, January 2006.

[71] Z. Qi, H. Li, S.X.-D. Tan, Y. Cai, and X. Hong. On-chip decoupling capacitor budgeting by sequence of linear programming. In *6th International Conference On ASIC*, volume 1, pages 98 –101, October 2005.

[72] R. J. Ribando and K. Skadron. Many-core design from a thermal perspective. *Proceedings of the 2008 Design Automation Conference*, pages 746–749, 2008.

[73] J.A. Roy and I.L. Markov. Seeing the forest and the trees: Steiner wirelength optimization in placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(4):632 –644, April 2007.

[74] R. Sabelka and S. Selberherr. A finite element simulator for three-dimensional analysis of interconnect structures. *Microelectronics Journal*, 32(2):163 – 171, 2001.

[75] K. Sankaranarayanan et al. A case for thermal-aware floorplanning at the microarchitectural level. *Journal of Instruction-Level Parallelism*, 7:8–16, 2005.

[76] S. Sapatnekar, P. Zhou, and K. Sridharan. Power grid optimization in 3D circuits using MIM and CMOS decoupling capacitors. *IEEE Design & Test of Computers*, (99):1–1, 2009.

[77] T. Sato, H. Onodera, and M. Hashimoto. Successive pad assignment algorithm to optimize number and location of power supply pad using incremental matrix inversion. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 723 – 728, 2005.

[78] A. Schubert, R. Dudek, H. Walter, E. Jung, A. Gollhardt, B. Michel, and H. Reichl. Reliability assessment of flip-chip assemblies with lead-free solder joints. In *Proceedings of the 2002 Electronic Components and Technology Conference*, pages 1246–1255, 2002.

[79] C.-W. Sham and E.F.Y. Young. Routability-driven floorplanner with buffer block planning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(4):470 – 480, April 2003.

[80] K. Sheth, E. Sarto, and J. McGrath. The importance of adopting a package-aware chip design flow. In *Proceedings of the 2006 Design Automation Conference*, pages 853–856, 2006.

[81] J. Singh and S.S. Sapatnekar. Partition-based algorithm for power grid design using locality. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 664 – 677, April 2006.

[82] M.R. Stan et al. Hotspot: A dynamic compact thermal model at the processor-architecture level. *Microelectronics Journal*, pages 1153–1165, 2003.

[83] N. Strusevich, S. Stoyanov, D. Liu, C. Bailey, A. Richardson, N. Dumas, JM Yannou, and V. Georgel. Modelling the behavior of solder joints for wafer level SiP. In

Proceesings of the 2006 IEEE Electronics Packaging Technology Conference, pages 127–132, 2006.

[84] H. Su, S. S. Sapatnekar, and S. R. Nassif. An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts. In *Proceedings of the 2002 International Symposium on Physical design*, pages 68–73, 2002.

[85] E. Suhir, Y.C. Lee, and C.P. Wong. *Micro-and Opto-Electronic Materials and Structures: Physics, Mechanics, Design, Reliability, Packaging: Volume I Materials Physics-Materials Mechanics. Volume II Physical Design-Reliability and Packaging*, volume 1. Springer, 2007.

[86] X. Tang, R. Tian, and D.F. Wong. Fast evaluation of sequence pair in block placement by longest common subsequence computation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(12):1406–1413, December 2001.

[87] R. Thorpe, D.F. Baldwin, and L.P. McGovern. High throughput flip chip processing and reliability analysis using no-flow underfills. In *Electronic Components and Technology Conference*, pages 419 –425, 1999.

[88] K. Toshiki et al. Impact of self-heating in wire interconnection on timing. *IEICE Transactions on Electronics*, 93(3):388–392, March 2010.

[89] C.-H. Tsai and S.-M. Kang. Cell-level placement for improving substrate thermal distribution. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(2):253–266, 2000.

[90] H.-C. Tsai and W.-R. Jong. The significance of underfill on the IC packages subjected to temperature cyclic loading. *Journal of Reinforced Plastics and Composites*, 26(12):1211–1223, 2007.

[91] J.-L. Tsai et al. Temperature-aware placement for SOCs. *Proceedings of the IEEE*, 94(8):1502–1518, 2006.

[92] K.-N. Tu. *Solder Joint Technology: Materials, Properties, and Reliability*. Springer, New York, 1st, edition, 2007.

[93] R. Vattikonda, W. Wang, and Y. Cao. Modeling and minimization of PMOS NBTI effect for robust nanometer design. In *Proceedings of the 43rd annual Design Automation Conference*, pages 1047–1052, 2006.

[94] B. Wang and P. Mazumder. Fast thermal analysis for VLSI circuits via semi-analytical green's function in multi-layer materials. *Proceedings of the 2004 International Symposium on Circuits and Systems*, 2:409–412, 2004.

[95] C.-Y. Wang and W.-K. Mak. Signal skew aware floorplanning and bumper signal assignment technique for flip-chip. In *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, pages 341–346, January 2009.

[96] T.-Y. Wang and C. C.-P. Chen. Thermal-ADI: A linear-time chip-level dynamic thermal simulation algorithm based on alternating-direction-implicit (ADI) method. In *Proceedings of the 2001 International Symposium on Physical design*, pages 238–243, 2001.

[97] T.-Y. Wang and C. C.-P. Chen. Optimization of the power/ground network wire-sizing and spacing based on sequential network simplex algorithm. In *Proceedings of the 2002 IEEE Symposium on Quality of Electronic Design*, page 157, 2002.

[98] T.-Y. Wang and C. C.-P. Chen. Thermal-ADI: a linear-time chip-level dynamic thermal-simulation algorithm based on alternating-direction-implicit (ADI) method. *IEEE Transactions on Very Large Scale Integration Systems*, 11(4):691–700, August 2003.

[99] T.-Y. Wang, J.-L. Tsai, and C. C.-P. Chen. Thermal and power integrity based power/ground networks optimization. In *Proceedings of the conference on Design, Automation and Test in Europe*, 2004.

[100] E. Wong and S. K. Lim. 3D floorplanning with thermal vias. *Design Automation and Test in Europe Conference*, 1:188, 2006.

[101] E. Wong, J.R. Minz, and S. K. Lim. Decoupling-capacitor planning and sizing for noise and leakage reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(11):2023 –2034, November 2007.

[102] I. Koren Y. Han and C. A. Moritz. Temperature aware floorplanning. In *Workshop on Temperature Aware Computer Systems*, 2005.

[103] J.-T. Yan, K.-P. Lin, and Y.-H. Chen. Decoupling capacitance allocation in noise-aware floorplanning based on DBL representation. In *Proceeeding of the 2005 IEEE International Symposium on Circuits and Systems*, pages 2219 – 2222, May 2005.

[104] S. Yokogawa, H. Tsuchiya, and Y. Kakuhara. Effective thermal characteristics to

suppress joule heating impacts on electromigration in Cu/low-k interconnects. In *International Reliability Physics Symposium*, pages 717–723, May 2010.

[105] F.Y. Young, D.F. Wong, and H.H. Yang. Slicing floorplans with boundary constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9):1385–1389, 1999.

[106] Y. Zhan, Y. Feng, and S. S. Sapatnekar. A fixed-die floorplanning algorithm using an analytical approach. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference*, pages 771–776, 2006.

[107] Y. Zhan and S. S. Sapatnekar. Fast computation of the temperature distribution in VLSI chips using the discrete cosine transform and table look-up. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 87–92, 2005.

[108] Y. Zhan and S. S. Sapatnekar. High-efficiency green function-based thermal simulation algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(9):1661–1675, 2007.

[109] M. Zhao, Y. Fu, V. Zolotov, S. Sundareswaran, and R. Panda. Optimal placement of power supply pads and pins. In *Proceedings of the 2004 Design Automation Conference*, pages 165 – 170, 2004.

[110] M. Zhao, R. Panda, S. Sundareswaran, S. Yan, and Yuhong Fu. A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming. In *Proceedings of the 43rd annual Design Automation Conference*, pages 217–222, 2006.

[111] S. Zhao, K. Roy, and C.-K. Koh. Decoupling capacitance allocation for power supply noise suppression. In *Proceedings of the 2001 International Symposium on Physical Design*, pages 66–71, 2001.

[112] S. Zhao, K. Roy, and C.-K. Koh. Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(1):81–92, January 2002.

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

[114] P. Zhou et al. 3D-STAF: scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits. In *Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design*, pages 590–597, 2007.

[115] P. Zhou, K. Sridharan, and S.S. Sapatnekar. Optimizing decoupling capacitors in 3d circuits for power grid integrity. *IEEE Design & Test of Computers*, 26(5):15–25, 2009.