Performance and Temperature Aware Floorplanning Optimization for 2D and 3D Microarchitectures

A Thesis
Presented to
The Academic Faculty

by

Michael Healy

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Performance and Temperature Aware Floorplanning Optimization for 2D and 3D Microarchitectures

Approved by:

Asst. Professor Sung Kyu Lim, Advisor
School of Electrical and
Computer Engineering
Georgia Institute of Technology, Advisor

Asst. Professor Hsien-Hsin Sean Lee
School of Electrical and
Computer Engineering
Georgia Institute of Technology

Asst. Professor Gabriel H. Loh
College of Computing
Georgia Institute of Technology

Date Approved: April 10th, 2006

Dedicated to my family,

who have always put up with me.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Professor Sung Kyu Lim for his guidance of my research and his patience during my studies at Georgia Tech. I would like to thank my thesis committee members, Professor Hsien-Hsin Sean Lee and Prof. Gabriel H. Loh for their valuable suggestions. I would like to thank all the members of GTCAD and CREST groups for their support and friendship, especially, Mongkol Ekpanyapong, Jacob Minz, Eric Wong, Mohit Pathak, Ismail F. Baskaya, J.C. Park, Kiran Puttaswamy, Chinnakrishnan S. Ballapuram, and Mario Vittes who helped me in many ways throughout my years at Georgia Tech.

My family has always supported me in all my decisions and for that I express my deepest gratitude and love. I also would like to thank my roommate and friend Christopher Tillotson who put up with me during many stressful times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The main objective of this thesis is to develop a physical design tool that is capable of being used by microarchitects to evaluate the impact of their design decisions on the physical design aspects of future microprocessor development. For deep submicron technology wire delay will scale increasingly badly compared to gate delay and so will become a major bottleneck to performance improvement. Three dimensional integrated circuits (3D ICs) offer a new method of dealing with non-linear wire latency by allowing shorter interconnects that act within their linear region. Thermal considerations in 3D ICs will be more important than traditional designs however, so this problem must also be addressed.

This thesis presents a microarchitectural floorplanning tool that will help computer architects to attack the wire delay problem early in the design stages of higher performance microprocessors by including consideration of design for 3D ICs. Consideration of the new problems that occur due to the move to 3D and inherent difficulties with deep submicron design is included. Experiments demonstrate that this tool can generate microprocessor floorplans that include many objectives and continue to enhance performance into the next generation of high performance design.

CHAPTER I

INTRODUCTION

The technology advances projected by the International Roadmap on Semiconductor Technology (ITRS) [42] imply that coming microprocessors implemented in deep submicron technologies will consume fewer clock cycles on performing useful computation than on communicating data operands or exchanging control information. Meanwhile, deep submicron devices and interconnects continue to be increasingly impacted by power and thermal densities, thereby eroding performance gains, threatening overall circuit reliability, and raising the cost for cooling solutions. Recently, microarchitectural level floorplanning has drawn significant interests from both the computer architecture and EDA communities, [34, 11, 6, 21, 38]. The main motivation behind this interest is a concern over the ever-worsening wire delay problem of high-performance processors [1, 25]. The idea is that a collaborative effort between microarchitecture and physical CAD can overcome or at least mitigate these damaging effects.

Concurrently, 3D integrated circuits, the product of an emerging technology that vertically stacks multiple die with a die-to-die interconnect as illustrated in Figure 1, and related technologies have been rising in prominence and are now seen by many as the next major revolution in IC manufacture. The die-to-die via pitch of this technology is very small and provides the possibility of spreading circuits into the third dimension. This results in a decrease in the total wire length, which translates into smaller wire delay and less power dissipated due to those wires. This allows 3D ICs to utilize short and fast vertical routes on traditionally long and slow global interconnects to address the wire delay problem effectively and efficiently. Advances in 3D integration and packaging are undoubtedly gaining momentum and have become of critical interest to the semiconductor community. These 3D integrated circuit and package manufacturing technologies are rapidly being adopted by several leading companies for commercial applications.

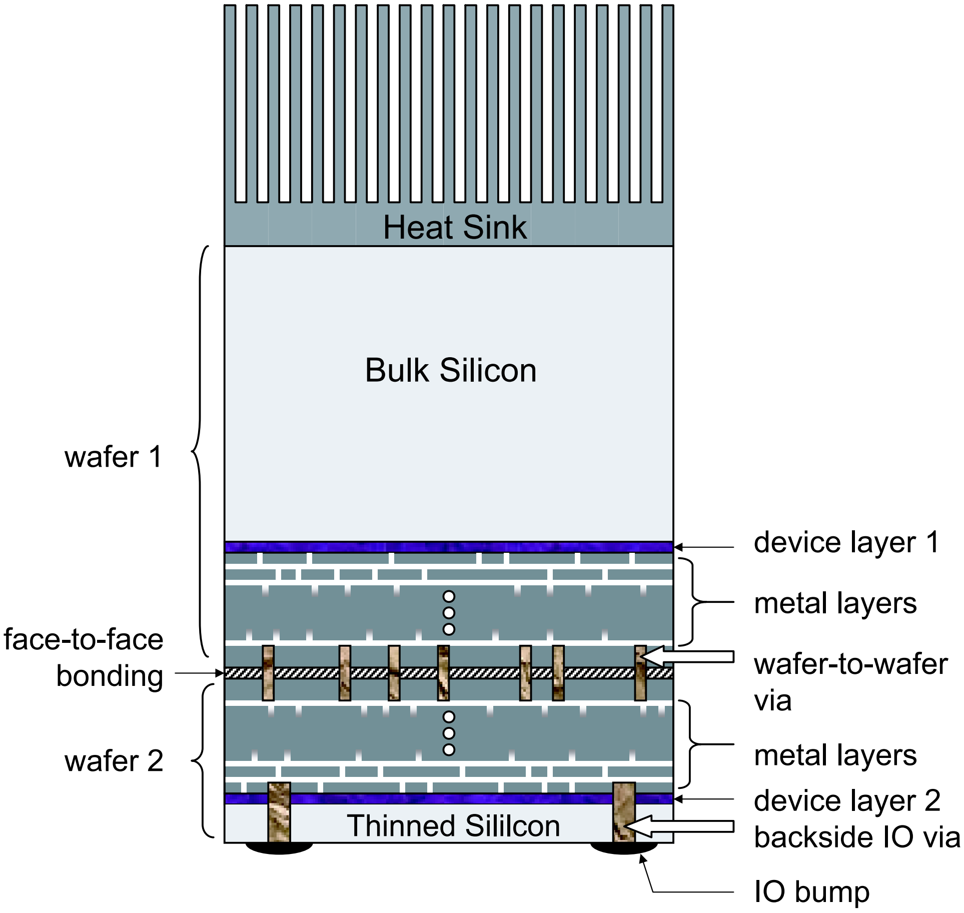


Figure 1: 2-die 3D IC with face-to-face bonding.

1.1 Problem Statement

There are many important metrics that are significantly impacted by the location of individual microarchitectural modules. First, the performance of a given microarchitecture (measured by IPC) is greatly influenced by floorplanning, as high target clock frequencies imply that global interconnects between modules are likely to be pipelined. Thus the access latency on all inter-module interconnects may increase or decrease dependent on the floorplan. Second, the floorplan is highly correlated to the thermal and leakage profile. This is because both the heat generation rate of each individual module and the thermal coupling between it and its neighbors affects that module's temperature. Moreover, the leakage power of each transistor is exponentially proportional to the temperature of that transistor. Finally, the dynamic power consumption of the buses and clock distribution network is affected by floorplanning. The total number of flip-flops (FFs) inserted on global interconnects changes the dynamic power consumed by the clock distribution network. It must be taken into account that shorter distance among the hot modules improves the performance while exacerbating the thermal issue. Thus the performance and temperature objectives conflict with each other. To address the different design constraints of different domains, a goal-directed, automated floorplanner that allows users to weight their own design requirements and make effective design trade-offs is required.

1.2 Contributions

The contributions of this thesis are as follows:

- It proposes the first multi-objective floorplanner for deep submicron processors at the *microarchitectural level*. To the best of this author's knowledge no other microarchitectural floorplanning for 3D ICs has even been investigated before. 2D/3D floorplanners simultaneously consider performance, thermal reliability, footprint area, and interconnect length objectives, providing various tradeoff points.
- It contains microarchitectural level thermal modeling that considers the temperature and leakage inter-dependence for effective thermal runaway avoidance. Also, the microarchitectural power analyzer, integrated with the thermal analyzer, models the dynamic and leakage power consumed by functional modules, global interconnects, and the clock distribution network for higher modeling accuracy.
- It provides an in-depth discussion of the bonding-style aware layer partitioning problem. Also, a discussion of how layer partitioning is done under different inter-die via requirements existing in face-to-face, face-to-back, and back-to-back bonding in 3D stacked ICs is given, as well as an examination of different layer paritioning algorithms.
- It presents a floorplanning optimizer that consists of two steps: initial solution construction via Linear Programming and stochastic refinement via Simulated Annealing. This hybrid approach proves to be very effective in obtaining high-quality solutions in short runtime.

1.3 Thesis Organization

This thesis is organized into seven chapters.

- **Chapter 1: INTRODUCTION** discusses the origin and history of the problem and presents motivations.
- **Chapter 2: RELATED WORK** presents a discussion of related works.
- **Chapter 3: SIMULATION INFRASTRUCTURE** presents the architecture model as well as the temperature and leakage simulators.
- **Chapter 4: 2D MICROARCHITECTRUAL FLOORPLANNING** presents the multi-objective 2D floorplanner.
- **Chapter 5: EXTENSION TO 3D FLOORPLANNING** discusses the 3D extension of the 2D floorplanner.
- **Chapter 6: EXPERIMENTAL RESULTS** shows details of the experiments that were run and their results.
- **Chapter 7: CONCLUSIONS** concludes this thesis.

CHAPTER II

RELATED WORK

There are several major divisions of related work. Many recent studies have focused on traditional 2D microarchitectural floorplanning for performance optimization but not thermal concerns [34, 11, 6, 21, 38]. For example, [38] uses a statistical design of experiments to approximate the effect on IPC of various wire lengths and then uses this approximation during simulated annealing based floorplanning to improve performance.

Several microarchitecture research works on temperature [43, 28, 5] and leakage power [16, 32, 17, 30, 24] provide runtime management of the functional modules but do not perform floorplanning. In [24], the most recently published, they present a system level leakage power model and discuss dynamic management to reduce the thermal problem, as well as discussing thermal runaway and showing that a dynamic management scheme must include consideration of leakage power to be effective.

Most existing floorplanning and placement works focusing on temperature [46, 10, 29, 39, 7, 3, 12] target circuit designs, not microarchitectural designs. For example, [12] presents a 3D temperature driven floorplanner based on TCG and a novel bucket structure to represent module overlap. They use various thermal analyzers to trade off runtime with accuracy and overall performance. A comparison betwen [12] and this work is given in Chapter 6 to demonstrate the general effectiveness of the approach.

Finally, recently developed physical design tools for 3D ICs [14, 23, 45, 49, 3, 36, 9, 22, 40, 31, 15, 12, 13, 8, 35] target gate-level netlists, are inefficient, and are not suitable for evaluating different microarchitecture options during the early design stage. Thus, this work is the first to simultaneously consider performance, temperature, and leakage for the automated floorplanning of an entire processor microarchitecture implemented on a 3D IC with full simulation of the results of floorplanning.

CHAPTER III

SIMULATION INFRASTRUCTURE

3.1 Microarchitectural Model

An illustration of the microarchitecture used in these experiments is provided in Figure 2. Each block represents a microarchitectural module used by the floorplanner. Each wire is isolated and modeled as a seperate resource that consumes power and has a delay in proportion to its length in order to model performance more faithfully for deep submicron processors. High frequency processors designed with deep submicron technologies will no longer be correctly simulated by architectural simulators that ignore inter-module communication latencies due to wire delays, floorplan constraints, and thermal concerns. Both performance evaluation and floorplanning must take into account the inter-module latency, which is a function of distance, and the number of flip-flops between modules. For this reason, the distances generated by the floorplanner are used to determine the latency-related parameters such as pipeline depth and communication/forwarding latencies for performance simulation.

While the algorithm presented here is general enough to consider virtually any configuration, a single one was chosen for the sake of expediency. The microarchitectural configuration used in this study is summarized as follows: the machine width is 8. The branch predictor is a 1024-entry gshare, the register update unit (RUU) [44], which combines the functionality of a reservation station and a reorder buffer, has 512-entries, the instruction and data L1 caches are 16KB, the unified L2 cache has 256KB and there is no L3 cache, there are 128-entry instruction and data TLBs, 8 ALUs, 4 FPUs, and a 64-entry load store queue.

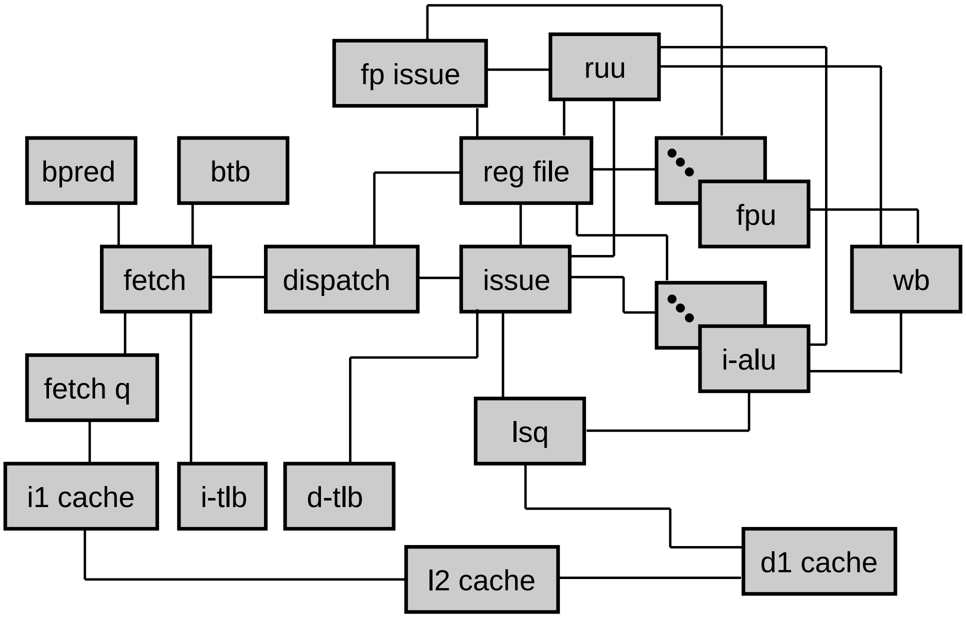


Figure 2: Processor microarchitecture model.

3.2 Intermodule Communication

It has previously been demonstrated, [21], that optimizing weighted wirelength based on the most highly used wires is a better hueristic for performance improvement than pure wirelength alone. Thus this work utilizes a cycle accurate simulation using SimpleScalar [2] to collect the intermodule traffic that occurs on each wire considered during floorplanning. Counters were added to collect totals and normalized weighted wirelength is used during the optimization phase.

3.3 Dynamic Power Modeling

The power consumption profile for each microarchitectural module is generated while the inter-module traffic is collected. It is gathered cumulatively for every hundred thousand cycles and then averaged over all samples. The rationale for this sampling is that the temperature is very unlikely to elevate abruptly within a processor's operation period of a few hundred thousand cycles due to the thermal time constants of the constituent materials of an IC. Note that only once at the very beginning of the entire design flow are these detailed traffic activity and dynamic power profiles collected. These power statistics are then used by the temperature analyzer to generate the temperature profile. A new floorplan is then created for the given temperature profile and module netlist by the floorplanner.

It is a major assumption of this work that the intra-module dynamic power consumption remains the same for different floorplans because the module activity factors primarily depend on the program behavior rather than the relative positions of the microarchitectural modules. However, since the new floorplan may lead to different interconnect lengths between modules, all of the inter-module interconnect powers are recomputed by the tool based on the new lengths and are added to the dynamic per-module power collected earlier.

Extremely high clock frequencies will require large numbers of flip-flops to be inserted on clock distribution network wires. This results in a large load on the distribution network. The increasing percentage of the power budget that the clock distribution network consumes combined with this fact necessitates modeling the clock power at a finer granularity. Therefore, the accurate clock power model from [18] is used in this work. Their model considers clock distribution network power for memory structure precharge arrays, distribution wiring and drivers, pipeline flip-flops, and the phase locked loop.

3.4 Leakage Power Modeling

The leakage power is modeled in a separate process within the design flow. The model considers different bias conditions, though it only estimates subthreshold leakage power, and is based on [47]. For array-like structures, such as caches and TLBs, the number of bits (or SRAM cells) stored is multiplied by the amount of leakage current per bit and by the supply voltage to calculate the total leakage power for the structure. To calibrate the model used here, the subthreshold leakage currents were also calculated using the method in eCACTI [20]. This model closely matches the leakage power estimated from eCACTI. For logic structures, CMOS gates are assumed where half the transistors are leaking at any given time. The area values from GENESYS [19] are used in estimating the number of transistors in these structures.

The following equation shows the relationship between the subthreshold leakage current

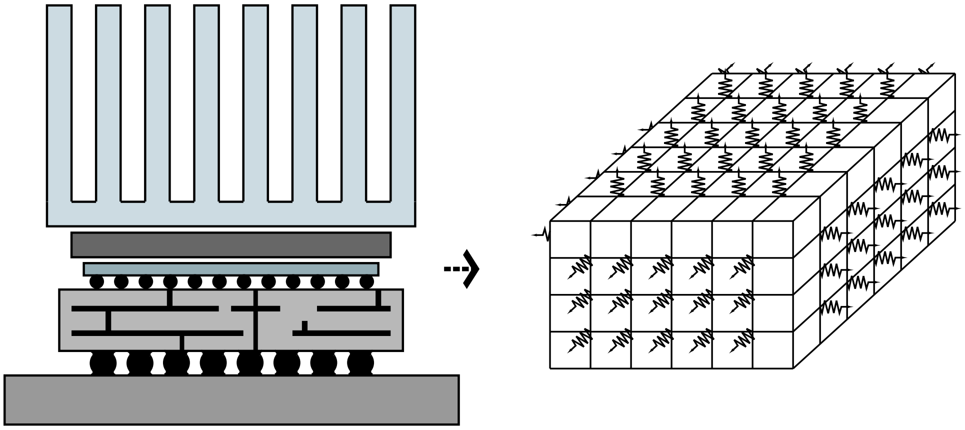


Figure 3: 3D grid of a chip for thermal modeling

I_{sub} and a given temperature θ:

$$I_{sub} = k \cdot W \cdot e^{-V_{th}/nV_\theta}(1 - e^{-V_{dd}/V_\theta})$$

where k and n are experimentally derived, W is the gate width, V_{th} is the threshold voltage, and V_{dd} is the supply voltage. V_θ is the thermal voltage that increases linearly as temperature elevates. Due to the temperature dependence on the subthreshold leakage current, leakage power is first estimated based on an initial temperature. The results are then fed to the thermal analyzer so that it can estimate the temperature and the leakage power more accurately. This is done within the thermal analyzer to model their interdependence. First a static leakage estimation is used to calculate the baseline temperature, then a leakage power based on those temperatures is calculated, then a new temperature based on the previous iterations leakage power, and so on until convergence or thermal runaway is detected. The criteria from [33] are followed for detecting thermal runaway: (i) the maximum module temperature T_{max} is increasing, or (ii) the positive change in power is larger than the package's heat removal ability. The package's heat removal ability is defined as $(T_{max} - T_a)/R_t$, where T_a and R_t are ambient temperature and thermal resistance of the package.

3.5 Thermal Modeling

The thermal model used in this work is based on the linearized differential equation ($k \cdot \nabla^2 T + P = 0$) for steady state heat flow, as described in [46]. In the equation, T is the temperature, k is the thermal conductivity, and P is the power density of heat sources. The chip is divided into a 3D grid as shown in Figure 3 to apply a finite difference approximation to the differential equation. The thermal equation is rewritten into the following matrix form: $\mathbf{R} \cdot \overrightarrow{P} = \overrightarrow{T}$, where $\overrightarrow{P}$ is the power profile vector, $\mathbf{R}$ is the thermal resistance matrix ($\mathbf{R}_{i,j}$ is the thermal resistance between node i and node j), ($\overrightarrow{P}_i$ is the power dissipation of node i), and $\overrightarrow{T}$ is the temperature profile vector ($\overrightarrow{T}_i$ is the temperature of node i). Thus, a single matrix-vector multiplication can be used to calculate the temperature of all active nodes using the power. The bus power for each net is added to the total power of the source block. The clock power is distributed evenly across the modules according to their areas. Then, the temperature and leakage power of each module are calculated iteratively using the thermal model until they either converge or thermal runaway is detected. The average number of iterations needed was found to be approximately 7 for the largest number of layers. A smaller number of layers requires fewer iterations.

A non-uniform 3D thermal resistor mesh, where grid lines are defined at the center of each microarchitectural module, was used in order to facilitate fast but reasonably accurate temperature calculation. The grid lines are defined for the Y and X directions and extend through the Z direction to form planes. The intersection of grid lines in the X and Y directions define the thermal nodes of the resistor mesh. Each thermal node models a rectangular prism of silicon that may dissipate power if it covers some portion of a block. The total power of each block is distributed according to and among the X-Y area of the nodes that block covers.

This thermal model is designed to provide fidelity for the optimization process, not accuracy. In order to provide a comparison with existing tools Table 1 is provided, a comparison of the temperatures calculated by the non-uniform model and those provided by Hotspot v3.0 [27] across ten benchmarks on a single floorplan. One can observe from the table that the non-uniform model provides a similar temperature and provides fidelity between various power profiles.

Table 1: Comparison with Hotspot v3.0 [27]. Temperatures are in $^\circ C$.

SPEC bench	equake	mcf	swim	art	gap	gzip	bzip2	vpr	twolf	lucas
Hotspot	68.0	69.0	72.8	84.3	92.6	119.3	125.7	149.1	164.6	174.7
Our Analyzer	86.2	86.6	88.0	92.0	94.5	104.7	107.4	114.4	118.9	123.0

3.6 Integrated Design Flow

The design flow used in this thesis incorporates the dynamic power, leakage power, performance, and thermal analysis discussed earlier into the floorplanner. Figure 4 illustrates an overview of this design flow. First, technology parameters and an architectural description are used to estimate the area and delay of the microarchitectural modules. The following analytical tools are used: CACTI [41] and GENESYS [19]. Then a cycle accurate simulation using SimpleScalar [2] combined with Wattch [4] is done in order to estimate dynamic power consumption for each benchmark and collect and extract the amount of traffic between modules. From these tools a profile weighted module netlist and power consumption information are extracted and then fed into the multi-objective floorplanner. The clock power estimation from [18] and the leakage estimation from [47] as described above are also integrated with the thermal analyzer.

The floorplanner proposed by this thesis consists of two steps: initial solution construction via Linear Programming (LP) and stochastic refinement via Simulated Annealing (SA). The floorplan area is recursively bipartitioned until each module is confined in its own partition. Each bipartitioning solution is optimized by a linear program where performance and thermal objectives are simultaneously considered under the leakage power constraint. The temperature and leakage profile is updated by a call to the thermal/leakage analyzer upon each bipartitioning. The interdependence between leakage power and temperature creates the possibility of thermal runaway [24], in which the temperature and leakage are caught in a positive feedback loop, which will cause device failure. If the floorplanner decides that thermal runaway is unavoidable given the current clock frequency then it scales the frequency down until it succeeds in avoiding runaway. The current solution is further optimized during SA-based refinement once the recursive bipartitioning is finished. Low-temperature

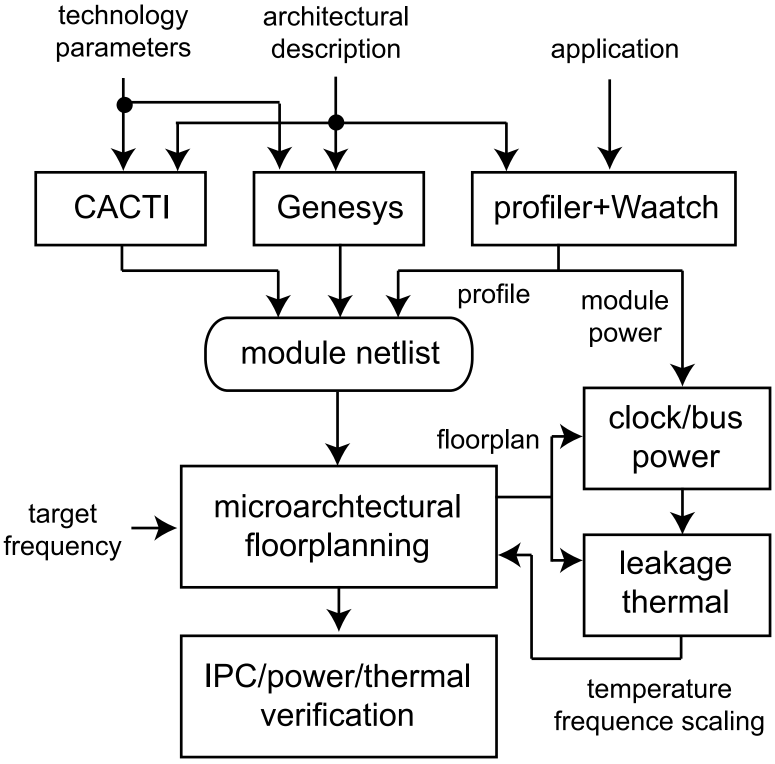


Figure 4: Overview of the microarchitectural floorplanning framework.

annealing is performed to fine-tune the LP-based solution where the optimization is again guided by the thermal/leakage analyzer. When the final solution is obtained, SimpleScalar, Wattch, and the thermal/leakage analyzer are used to evaluate the final solution for IPC, power, and thermal metrics.

CHAPTER IV

2D MICROARCHITECTURAL FLOORPLANNING

Given the area of a set of microarchitectural modules and a netlist that specifies the connectivity among these modules, the multi-objective 2D microarchitectural floorplanner tries to determine the width and height of each module and to place it into a single chip such that (i) there is no overlap among modules, (ii) a user-specified clock frequency constraint is satisfied, and (iii) thermal runaway does not occur under that constraint. The goal is to provide a floorplan that effectively maximizes the performance of a processor while simultaneously minimizing the footprint area of the floorplan and maximum module temperature for better thermal reliability. The LP-based floorplan construction and simulated annealing-based refinement used in this work are discussed in this chapter.

4.1 LP-based 2D Floorplanning

The slicing floorplanning algorithm is described in Figure 5. The basic idea behind the algorithm is to perform recursive bipartitioning until each partition contains a single module as shown in Figure 6. In this approach the overall relative location among the modules is determinded by the slicing operation, while an LP determines the dimension of the modules and fine-tunes the location. After a partition to be divided is chosen module temperatures are obtained by performing thermal/leakage analysis. Because there is no way to obtain block temperatures without a floorplan the first iteration of the recursive bi-partitioning contains no temperature objective. From then on the previous iteration's block positions are used to calculate the temperatures for the current iteration. LP-based floorplanning is then used to simultaneously optimize the performance and temperature distribution under the target frequency, leakage, center of gravity constraints (to remove overlap among the modules), and boundary constraints. An *iteration* in this algorithm combines a single bipartitioning and a subsequent LP-based floorplanning of all modules.

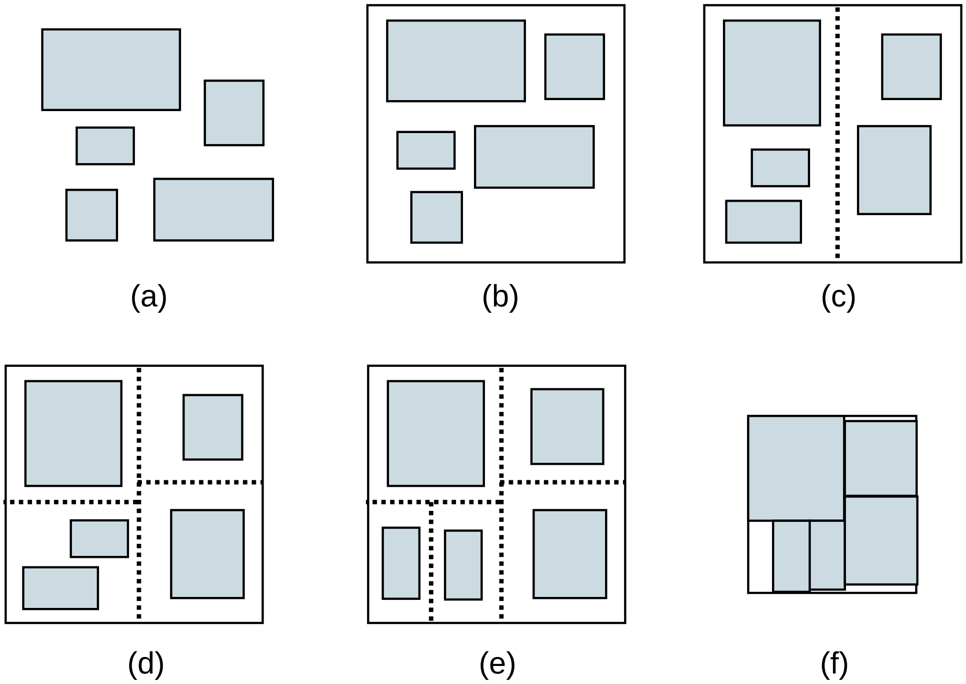
```
Performance and Thermal-aware Floorplanning
while (there exists a partition with multiple modules)
    Choose a partition $j$ to be divided;
    Call thermal/leakage analysis;
    for (number_of_repetitions)
        Insert a cutline and compute center of gravity;
        Solve LP with inserted cutline;
    Pick the best cutline from the set of repetitions;
    Update centers of gravity and bounding boxes;
return $x_i$, $y_i$, $w_i$, $h_i$, $z_{ij}$ for all modules;
```

Figure 5: Description of the floorplanning algorithm. Top-down recursive bipartitioning and LP-based floorplanning solutions are obtained at each iteration.

Thus, $k-1$ iterations are performed if there are k modules in the netlist. Note that different cutlines can be obtained by multiple repetitions of each iteration. This will be done because there exist multiple solutions that satisfy the boundary and center of gravity constraints during each bipartitioning. Thus, each bipartitioning is performed several times and the best solution in terms of performance and temperature profile is picked.

The following variables are used for the LP-based floorplanning formulation:

- N: set of all modules in the netlist.
- E: set of all nets in the netlist.
- x_i, y_i: location of module i.
- w_i, h_i: half width and half height of module i
- a_i, g_i: area and delay of module i
- $w_m(i)$, $w_x(i)$: minimum/maximum width of module i
- $\lambda_{i,j}$: normalized profile weight on wire (i,j)
- $z_{i,j}$: number of flip-flops on wire (i,j) after insertion
- $X_{i,j} = |x_i - x_j|$ and $Y_{i,j} = |y_i - y_j|$
- $T_{i,j}$: normalized product of the temperature of modules i and j


(a) (b) (c)

(d) (e) (f)

Figure 6: Illustration of 2D microarchitectural floorplanning. (b-e) LP-based slicing floorplan, (f) non-slicing floorplan refinement.

- A: aspect ratio of the chip
- X_x : maximum x_i, Y_x: maximum y_i
- C: target cycle time
- d_r: unit length delay of repeated interconnects

The LP floorplanner determines the values for the following decision variables: x_i, y_i, w_i, h_i, and z_{ij}. The following are the variables used for bipartitioning:

- $B(u)$: set of all modules at iteration u
- $M_j(u)$: set of all modules in partition j at iteration u
- $S_{j,k}(u)$: set of modules assigned to subpartition k ($k \in \{1, 2\}$ for bipartitioning) in partition j at iteration u
- $(\bar{x}_{jk}, \bar{y}_{jk})$: center of subpartition k contained in partition j
- r_j,v_j,t_j,b_j: the right, left, top, and bottom boundaries of partition j

The LP formulation is used to perform floorplanning at iteration u of the main algorithm shown in Figure 5. The LP-based slicing floorplanning is formulated as follows:

Minimize:

$$\sum_{(i,j)\in E} (\alpha \cdot \lambda_{ij} \cdot z_{ij} + \beta \cdot (1 - T_{ij})(X_{ij} + Y_{ij}) + \gamma \cdot X_x) \tag{1}$$

Subject to:

$$z_{ij} \geq \frac{g_i + d_r(X_{ij} + Y_{ij})}{C},\ (i,j) \in E \tag{2}$$

$$X_{ij} \geq x_i - x_j \text{ and } X_{ij} \geq x_j - x_i,\ (i,j) \in E \tag{3}$$

$$Y_{ij} \geq y_i - y_j \text{ and } Y_{ij} \geq y_j - y_i,\ (i,j) \in E \tag{4}$$

$$z_{ij} \geq 0,\ (i,j) \in E \tag{5}$$

$$w_m(i) \leq w_i \leq w_x(i),\ i \in N \tag{6}$$

$$x_i,\ y_i \geq 0,\ i \in N \tag{7}$$

$$X_x \geq x_i \text{ and } A \cdot X_x \geq y_i,\ i \in N \tag{8}$$

Boundary Constraints:

$$\begin{align} x_i + w_i &\leq r_j,\ i \in M_j(u), j \in B(u) \tag{9} \\ x_i - w_i &\geq v_j,\ i \in M_j(u), j \in B(u) \tag{10} \\ y_i + m_i w_i + k_i &\leq t_j,\ i \in M_j(u), j \in B(u) \tag{11} \\ y_i - m_i w_i - k_i &\geq b_j,\ i \in M_j(u), j \in B(u) \tag{12} \end{align}$$

Center of Gravity Constrains: for $k \in \{1, 2\}, j \in B(u)$

$$\begin{align} \sum_{i\in S_{jk}(u)} a_i x_i &= \sum_{i\in S_{jk}(u)} a_i \times \bar{x}_{jk} \tag{13} \\ \sum_{i\in S_{jk}(u)} a_i y_i &= \sum_{i\in S_{jk}(u)} a_i \times \bar{y}_{jk} \tag{14} \end{align}$$

There are three terms in the objective function shown in Equation (1): profile-weighted wirelength ($= \lambda_{ij} \cdot z_{ij}$), thermal-weighted wirelength ($= (1 - T_{ij})(X_{ij} + Y_{ij})$), and footprint area ($= X_x$), where λ_{ij} is the profiled activity factor of the wire between modules i and j.

The minimization of the first term improves IPC while the minimization of the second term stretches the distance of two modules, thereby reducing thermal coupling. $(1-T_{ij})(X_{ij}+Y_{ij})$ was chosen as the temperature dependant portion of the cost function because it satisfies several properties: It is linear with respect to distance between module i and module j, it considers the temperatures of both module i and module j, and it grows smaller when considering hot blocks and larger when considering cool blocks. Because the cost function is being minimized in the LP and not maximized, it is necessary to only consider minimization of the distance between cool blocks and not maximization of the distance between hot blocks, as would be preferable. Since minimizing $X_x \cdot Y_x$ (= floorplan area) is non-linear, X_x is minimized since the constraint (8) enforces $A \cdot X_x$ to be greater than all y values. Note that α, β, and γ are user-defined parameters for weighing the performance, thermal, and area objectives. In case $\alpha = 0$, the floorplanner optimizes temperature+area only. In case $\beta = 0$, the floorplanner optimizes performance+area objective only. Lastly, the conventional area/wirelength-driven floorplanner uses the following new objective function:

$$\gamma \cdot X_x + \delta \cdot \sum_{(i,j) \in E} (X_{ij} + Y_{ij}) \tag{15}$$

An extensive comparison among these four different floorplanning objectives (simultaneous performance+temperature+area, performance+area, temperature+area, and area+wirelength) is given in Chapter 6. Note that the area objective is used in all of these variations. The area objective has a positive impact on performance and wirelength objectives and a negative impact on thermal objective.

The definition of latency is used to obtain constraint (2). If there is no FF on a wire (i, j), the delay of this wire is calculated as $d(i, j) = d_r(X_{ij} + Y_{ij})$. Then, $g_i + d(i, j)$ represents the latency of module i accessing module j, where $d(i, j)$ denotes the delay between i and j. Since C denotes the clock period constraint, $(g_i + d(i, j))/C$ denotes the minimum number of FFs required on (i, j) in order to satisfy C. Absolute values on x and y distance are given in (3)–(4). Constraint (5) requires that the number of FFs on each edge is non-negative. The block boundary constraints (9)–(12) require that all modules in the block be enclosed by these block boundaries. The center of gravity constraints (13)–(14) require that the module area-weighted mean (= center of gravity) among all modules in each sub-block corresponds to the center of the sub-block.

4.2 Stochastic Refinement

There are several non-optimalities introduced by the standard LP relaxation of the floorplanning problem. The recursive bipartitioning process also yields only slicing floorplans. In order to address these issues a simulated annealing based refinement engine was implemented for the floorplanner. This refiner searches around the local space to find a local minimum unconstrained by linearity. It is assumed that the LP floorplanner is able to come close to the true optimal and so doing local search is enough to close in on that point. There are three intralayer moves used during the simulated annealing refinement: Swapping in positive sequence, swapping in both positive and negative sequences, and rotation. These moves do not effect the floorplan enough to change the thermal model because they do not change the footprint of the floorplan a great deal, especially when searching through the local solution space. Moving modules on a static footprint would only change P, the location of the power dissipating sites. Any large changes to the footprint are modeled by scaling the floorplan to fit within the statically defined resistive mesh and appropriately scaling the resultant temperature due to higher power density. This was found to occur very infrequently during the low temperature annealing process. A sequence pair is derived from the LP floorplanning result and low temperature annealing is performed on them. The *gridding* scheme described in [37] is used to derive the corresponding sequence pair representation from the slicing floorplan. Specifically, the positive and negative loci are drawn for each module and then ordered to obtain the sequence pair. Next the initial annealing temperature is computed by setting the probability of accepting bad moves to a low value. This reduces the runtime required for the annealing process significantly and focuses on results that are near the LP based solution, which is assumed to be fairly close to optimal. The following cost function is used during annealing:

$$cost = \alpha \cdot perf\_wire + \beta \cdot max\_temp + \gamma \cdot area$$

where $perf\_wire$ is the profile-weighted wirelength and max_temp is the maximum module temperature. The same weighting constants, α and β from Equation (1), are used. It is important, however, to note that temperature is *not* the weighted distance between two hot blocks but the *actual* temperature obtained from the thermal analyzer. Thus, the thermal analysis is the runtime bottleneck during refinement since the analysis for potentially many candidate solutions must be performed during the annealing process. Consideration of performance is done in both the Simulated Annealing and Linear Programming approaches by the inclusion of profile weighted wirelength in the cost function. The authors in [12] suggest that only the vertical heat flow should be considered for fast approximation since the heat sinks are located on the top and bottom of the 3D structure in general. In this approach, however, all-direction thermal analysis is still performed but with a relatively coarse uniform resistive mesh.

Assuming that the thermal conductivity of functional modules are similar (they are composed mostly of silicon), swapping the location of modules would not change the thermal resistance matrix $\mathbf{R}$. Thus matrix $\mathbf{R}$ only needs to be computed once at the beginning. The power vector $\overrightarrow{P}$ is then updated and then multiplied by $\mathbf{R}$ to obtain the new temperatures. Alternatively, a change in power profile $\Delta\overrightarrow{P}$ can be defined. Multiplying $\mathbf{R}$ and $\Delta\overrightarrow{P}$ will give change in temperature vector $\Delta\overrightarrow{T}$. Adding $\Delta\overrightarrow{T}$ to the old temperature vector will give the new temperature profile. The power profile is usually only minimally affected by swapping two blocks, so $\Delta\overrightarrow{P}$ is usually sparse. The number of multiplications required by the second method is then reduced at the expense of doing extra additions and subtractions. Lastly, the leakage and clock power updates are done faster since it basically involves evaluating a set of equations based on the new module locations and temperature values.

CHAPTER V

EXTENSION TO 3D FLOORPLANNING

A new approach in floorplanning as well as updates on the architectural simulation for performance, power, and thermal evaluation are required for the extension to 3D ICs. The 3D floorplanning algorithm considers the issues that are specific to 3D: bonding-aware layer partitioning. This problem is solved using the LP-based 3D slicing floorplanning plus stochastic non-slicing floorplan refinement.

5.1 3D Extension of Architectural Simulation

The simulation engines discussed in Chapter 3 are extended in order to support the performance, power, and thermal simulation for 3D microarchitecture floorplanning as follows:

- Performance: the simulation of benchmarks for the IPC computation in the 3D is not much different from that for the 2D case except that the access latency on each interconnect is calculated based on a 3D floorplan that involves delay in the z-dimension.
- Dynamic power: again it is assumed that module power is independent of floorplanning. However, bus and clock power are heavily dependant on floorplanning, which will give a reduction in interconnect lengths for a 3D floorplan. The existing bus power calculator is extended to consider inter-layer interconnects. Again H-trees are used for each layer, and these H-trees are connected by through-vias. The number of FFs and buffers included in the 3D clock tree is calculated based on the area of each layer.
- Temperature/leakage: the temperature calculation for 3D becomes more complextue to the multiple die structure. Thus, more layers are added in the 3D mesh to model the multiple sets of device, metal, and bonding layers. The leakage power computation is straightforward, simply based on an equation, once the temperature for each module

is known.

Finally, the architecture-to-floorplan design flow shown in Figure 4 remains the same except that all related boxes are now are 3D-aware.

5.2 Bonding-aware Layer Partitioning

3D ICs require special kinds of vias for inter-die connections called *through-vias*. There are three kinds of through-vias depending on the style of bonding mechanism used to bond two die together: *face-to-face* (F2F), *face-to-back* (F2B), and *back-to-back* (B2B) through-vias, as illustrated in Figure 7. The "face" refers to the metal layer side of a die, whereas the substrate side is called the "back". F2F through-vias ($\approx 0.5\mu \times 0.5\mu$) have a smaller pitch than F2B ($\approx 5\mu \times 5\mu$) and B2B through-vias ($\approx 15\mu \times 15\mu$) [26]. In addition, too many F2B/B2B through-vias fabricated on a single thinned wafer may adversely affect its reliability [48] since these vias actually penetrate the substrate. Thus, it is desirable to *reduce* the number of inter-die connections in F2B/B2B bonding. In the case of F2F bonding, however, it is desirable to *increase* the number of inter-die connections since the via density is much higher (almost the same as intra-die via density) and thus enables a significantly higher bandwidth for inter-layer communication. Note that F2B/B2B bonding is inevitable if the number of die exceeds two. Moreover, in the case that more than one bonding style is used in a single 3D IC, the 3D floorplanning has to be done carefully to exploit both bonding styles.

In the two-step approach for 3D floorplanning, the modules are first partitioned into layers (= die) and then these layers are floorplanned simultaneously. The goal during layer partitioning is to exploit the bonding style of the manufacturing process and vertical overlap opportunities, whereas floorplanning optimizes the vertical overlap for performance, footprint area, and thermal objectives. During layer partitioning, a layer is assigned to each module such that the connection at the F2F boundary is maximized while the F2B/B2B connection is minimized. Next, the pair of modules connected via high profile-weighted edges is split into two layers with F2F bonding so that they can be vertically overlapped during the subsequent floorplanning steps to achieve better performance. In addition, highly

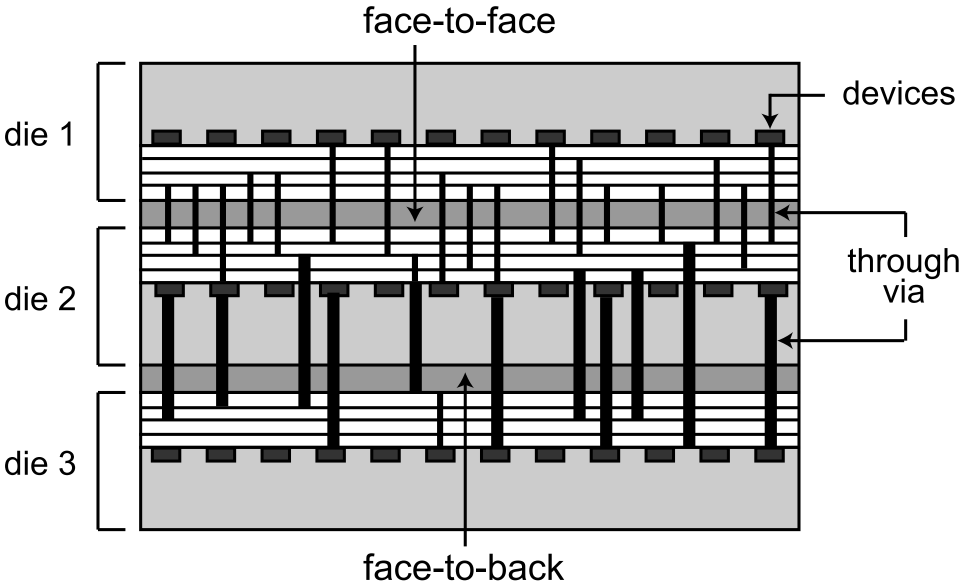


Figure 7: Through vias in 3D ICs with face-to-face and face-to-back bonding. Back-to-back style forms when the two substrate sides are attached (not shown in this figure).

active modules are split in the same way, i.e., two layers with F2F bonding, such that the shorter interconnect that bridges these modules help reduce the overall power consumption of the busses. Since the temperature of the modules requires floorplanning, the layer partitioning is not temperate-aware. Finally, the modules with large area, such as the RUU, are seperated into different layers to help minimize the footprint area and reduce the amount of white space. In the area greedy construction algorithm, the modules are sorted according to their size, power density, and switching activity. The best possible layer for each module is then assigned based on the performance, power, and area objectives mentioned earlier.

During the LP bipartitioning and floorplanning it is difficult to allow modules to switch layers. This is why a static layer assignment is done prior to any floorplanning. In order to examine the effect of partitioning several partitioning algorithms were chosen and then run. Experimentation demonstrated that area distribution between layers was extremely important for generating reasonable footprints. Partitioners such as greedy area based, bonding style aware with equal area distribution, as described above, and profile weighted bonding aware paritioning were all examined. Each partitioning style was run in parallel and the option with the lowest cost was chosen. In all cases the bonding style aware with equal area distribution outperformed the rest due to the combination of wirelength and footprint area. An analysis of this is presented in Chapter 6.

5.3 LP-based 3D Floorplanning

In LP-based 3D floorplanning, the slicing floorplanning discussed in Section 4.1 is extended to handle multiple layers simultaneously. Specifically, each slicing cutline is inserted to cut all layers simultaneously as illustrated in Figure 8. The goal of slicing 3D floorplanning remains the same as the 2D case: to determine the dimension and relative position among the modules so that the multi-objective function is minimized. In addition, these locations are refined via the 3D non-slicing floorplanning during post refinement. The major difference between the 2D and 3D slicing floorplan is the interaction with different layers, which is a key element for an effective 3D floorplan. More specifically, area optimization has to be footprint-aware: the area increase from the smallest layer can be easily tolerated since it is less likely to increase the overall footprint area. The LP formulation reflects this new optimization goal that is unique to 3D floorplanning. Since layer partitioning has already addressed the bonding-style-related issues, the modules are not allowed to move to other layers during the floorplanning. This is partly based on the limitations of the linear objective function which is required during LP optimization.

The following 3D-related LP variables were used in conjunction with the 2D-related variables shown in Section 4.1: l_i: layer of module i, $L_{ij} = |l_i - l_j|$, d_v: delay of inter-layer vias. It is crucial to note that the LP objective function used for 2D floorplanning, i.e., Equation (1), can be used *as is* so long as *all* layers are considered simultaneously. Specifically, the $\alpha \cdot \lambda_{ij} \cdot z_{ij}$ term in Equation (1) minimizes the distance between the frequently communicating modules if these are in the same layer—if not, the vertical overlap will be maximized as long as the reference point of module location is consistent. The lower left corner of each module is used for this work. In addition, the $\beta \cdot (1 - T_{ij})(X_{ij} + Y_{ij})$ term separates two hot modules in the same layer *and* minimizes the vertical overlap between two hot modules in different layers. Finally, the $\gamma \cdot X_x$ term still captures the minimization of 3D footprint area as long as the X_x and Y_x are computed based on the modules in all

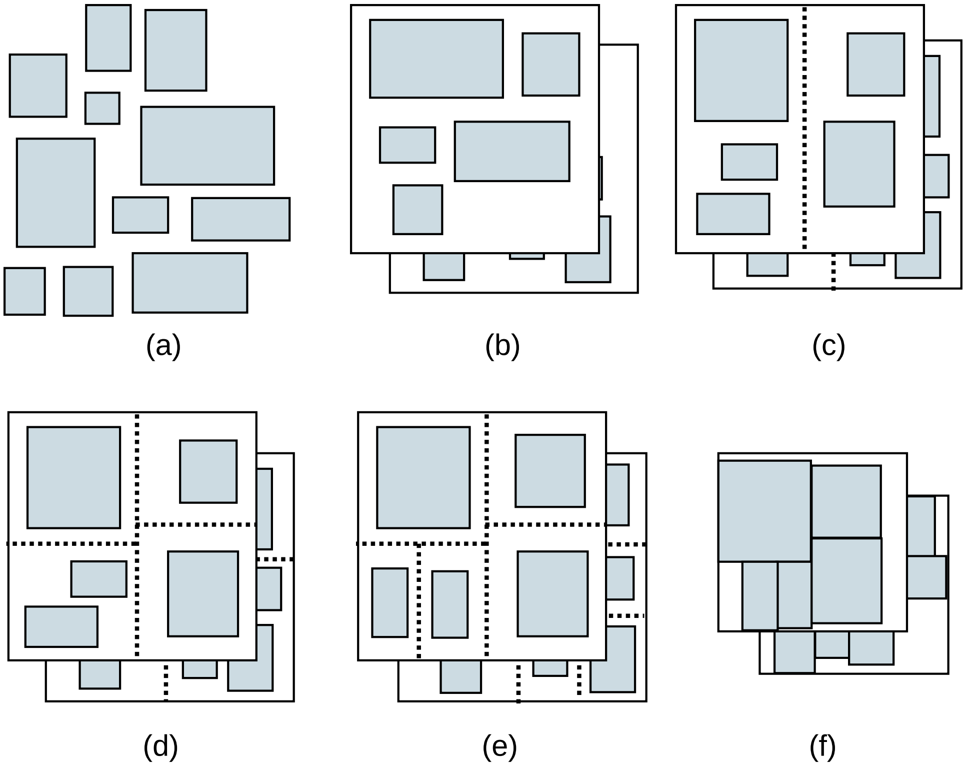


Figure 8: Illustration of 3D microarchitectural floorplanning. (b) layer partitioning, (c-e) LP-based 3D slicing floorplan, (f) non-slicing floorplan refinement.

layers. The only difference between the LP formulations of 2D and 3D floorplanning is the latency constraint, for which Equation (2) is updated with the following:

$$z_{ij} \geq \frac{g_i + d_r(X_{ij} + Y_{ij}) + d_v L_{ij}}{C}, \ (i,j) \in E \tag{16}$$

The delay of inter-layer vias as well as interconnect delays are considered by this delay constraint during the computation of the number of FFs needed to satisfy the clock period constraint C. It is assumed that d_r (= unit length delay of repeated interconnects) is larger than d_v (= delay of inter-layer vias).

5.4 3D Stochastic Refinement

The goal of the 3D stochastic refinement is to improve the 3D slicing floorplanning solution obtained from the LP-based construction algorithm. The basic approach is the same as the 2D case discussed in Section 4.2: non-slicing floorplanning with low-temperature simulated annealing to simultaneously refine the performance, temperature, and area objectives. The major difference between the 2D and the 3D case is that one sequence pair per layer is used to represent the entire 3D solution. In addition, the perturbation scheme does not allow inter-layer module movement in order to maintain the bonding-aware layer separation and remain close to the minima found by the LP solver. Finally, the temperature calculation takes even longer since the thermal model needs to be expanded to consider multiple die. Thus, the annealing schedule is adjusted so that the runtime is not increased too much. This involves tuning such parameters as the initial/final annealing temperature, total number of moves for each annealing temperature, cooling ratio, and the annealing termination criteria.

CHAPTER VI

EXPERIMENTAL RESULTS

Ten programs from the SPEC2000 benchmark suite were chosen to run our experiments on. We chose 4 from the floating point and 6 from the integer benchmark suites. For IPC evaluation, we ran each benchmark on the average case floorplan using a modified SimpleScalar 3.0 [2] by fast-forwarding 4 billion instructions and simulating the next 4 billion instructions. The reported temperature is simulated after all floorplanning steps and is adjusted relative to a $45°C$ ambient temperature. Our 3D floorplan is based on a 4-layer stacked IC. We assume face-to-face bonding between layer 0 (topmost) and 1 and between layer 2 and 3. A back-to-back bonding is used between layer 1 and 2. The heat sink is attached to layer 3. Wirelength is reported in mm. The "area" in our results refers to the footprint area (= maximum width × maximum height) of the 4-layer floorplan and is reported in mm^2. The runtime of our framework was collected on Pentium Xeon 2.4 GHz dual-processor systems. The runtime of profiling 4 billion instructions after fast-forwarding 4 billion instructions was about 4 hours per benchmark as was the power collection simulation for the same sets of instructions. The floorplanning steps took approximately 25 minutes and the simulations for the reported values of temperature and IPC took approximately 2 minutes and 1 hour per benchmark, respectively.

A comparison with [12] is given in Table 2. Our floorplanner was run with a combined area, wirelength, and temperature objective on the MCNC and one GSRC benchmark. One can observe from the table that our floorplanner had very comparable results while optimizing area more than wirlength. The weighting numbers used in [12] are unknown and so this accounts for the variation in these parameters.

Table 3 presents a comparison of the performance (P), temperature (T), area (A), wirelength (W), and runtime of 4 different objective functions for the 2D and 3D cases. All data in this table are taken from the combined LP+SA approach. A major assumption of

Table 2: Comparison with [12]. Our LP+SA floorplanner with an A+W+T objective is used. Our values are given as ratios with [12]'s.

	CBA-T [12]			LP+SA		
bench	area	wirelength	Temp($^{\circ}C$)	area	wirelength	Temp
ami33	4.14E+05	24442	160	0.94	1.11	0.96
ami49	1.84E+07	477646	151	0.79	1.21	0.94
n100	6.56E+04	92450	158	1.27	0.95	0.93

our work is that dynamic power dissipation does not change significantly due to changes in module position. Because we do not resimulate the dynamic power consumption during the evaluation phase all variations in temperature among the experiments are due to thermal coupling and changes to the clock power, bus power, and leakage power. One can see that for the 2D case the maximum module temperature increased markedly for A+P compared to the baseline A+W. The IPC result of A+P is the best among the 4 algorithms with an average IPC improvement over A+W by 35%. A+T decreases the temperature by about 24% over A+P while the IPC decreases by 25%. The hybrid A+P+T decreases the temperature by 14% over A+P while maintaining a high IPC value of 22% above the baseline A+W.

For the 3D cases, 3D A+W achieves a 37% increase in IPC and a 34% increase in temperature over 2D A+W while decreasing the total wirelength by almost 40%. The area result of 3D A+W is the best among all objective functions. A+P increases the IPC by 18% over A+W and increases the temperature by 19%. As expected A+T decreases the temperature result of A+P significantly and achieves the best temperature results among all four 3D algorithms. The 4X increase in grid size for the temperature simulations in the 3D case causes the runtime of those objectives incorporating temperature calculations to increase dramatically. The hybrid A+P+T retains a temperature close to that of A+W while increasing the IPC by 14%. In summary, A+P+T (i) obtains results that are between those of A+T and A+P and (ii) outperforms A+W in terms of performance with comparable temperature results for both 2D and 3D. In case the temperature should be more emphasized, the thermal weight can be increased, which will likely lead to performance degradation.

Also shown in Table 3 is the pipeline depth and whitespace percentages for the various objective functions. Pipeline depth is calculated by adding in the number of flipflops inserted between the major stages of the basic simplescalar pipeline. It is a rough approximation of what the actual pipeline depth would be to show that adding this number of flipflops is not unreasonable. Whitespace percentages for our floorplans are not phenomenal due to several factors. Most notably the large dispersion among the areas of our blocks, they vary by up to 3 orders of magnitude, causes area distribution to be an influential factor in the optimization effectiveness. Secondly temperature objectives can only be met by floorplanning alone by spreading out hot blocks. This can obviously lead to poor whitespace performance.

A tradeoff between performance and temperature is shown in Figure 9. Temperature and IPC are reported as averages over the 10 benchmarks. The performance and area weights are held constant while the thermal weight is varied. As expected the graph shows that as the thermal weight is given more consideration by the floorplanner the performance drops. Ideally there would be some separation between the curves to indicate that high reduction in temperature could occur with little degradation in IPC value. The sweet spot of the curve appears when the thermal weight is around 10. The IPC drops sharply after this and so would be undesirable for the reduction in temperature achieved. One can observe that there is a 15% reduction in IPC and a 22% reduction in temperature between the performance-only objective (0) and the highest weight hybrid objective (20) for the 3D case. As expected and also shown in Table 3 the multi-layer floorplans increase both the temperature and IPC over the single layer floorplans. Also of note is that the highest thermal weight multi-layer floorplan has a temperature close to that of the lowest thermal weight single layer floorplan while achieving a higher IPC. This demonstrates the benefits rendered by moving to multi-layer ICs.

Experimental results were also gathered across the three floorplanning algorithms; linear programming only, simulated annealing, and the combined approach of linear programming followed by simulated annealing refinement. Table 4 presents a comparison of the IPC, temperature, area, wirelength, and runtime of these three floorplanning algorithms for the 2D and 3D cases. One can observe for the 2D case from the table that the LP floorplanner

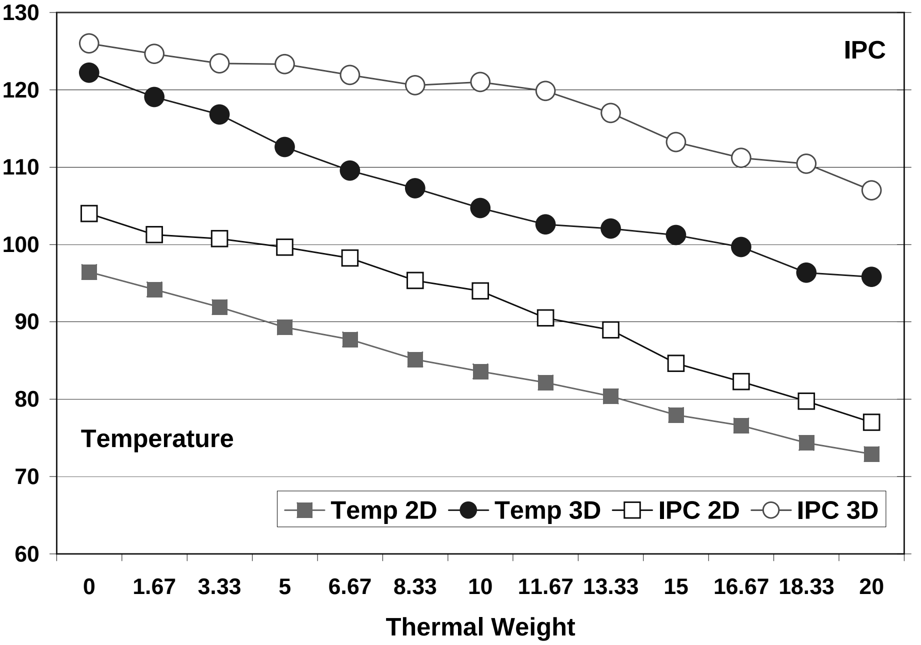


Figure 9: Tradeoff between performance and temperature. Performance and area weights are held constant while thermal weight varies.

does very poorly on the area of the floorplan and is not as good as the combined approach for IPC. The wirelength values are within an acceptable range for all approaches, though it is interesting to note that while the LP-only approach creates large area the wirelength values are actually less. This is because while wirelength was an objective during the recursive bipartitioning phase of the LP the area is not because the formulation has no way to constrain overlap. This was a large part of the motivation to use simulated annealing to refine the LP-only solution. In summary, LP+SA improves LP and outperforms SA consistently in terms of both performance and thermal objectives. The runtime of all approaches was roughly equivalent, showing that in a similar amount of time the combined approach produces better solution quality. These trends are consistent for the 3D cases with increased overall temperature averages and runtime. Again the large runtime increase was due mainly to the increase in simulation time for the temperature.

Figure 10 shows snapshots of our floorplanning solution. We use LP+SA with area, performance, and temperature objectives. The whitespace of the floorplan is somewhat less than optimal but this is due to the higher weights placed on the performance and temperature optimization. These floorplans also highlight the challenge in area optimization

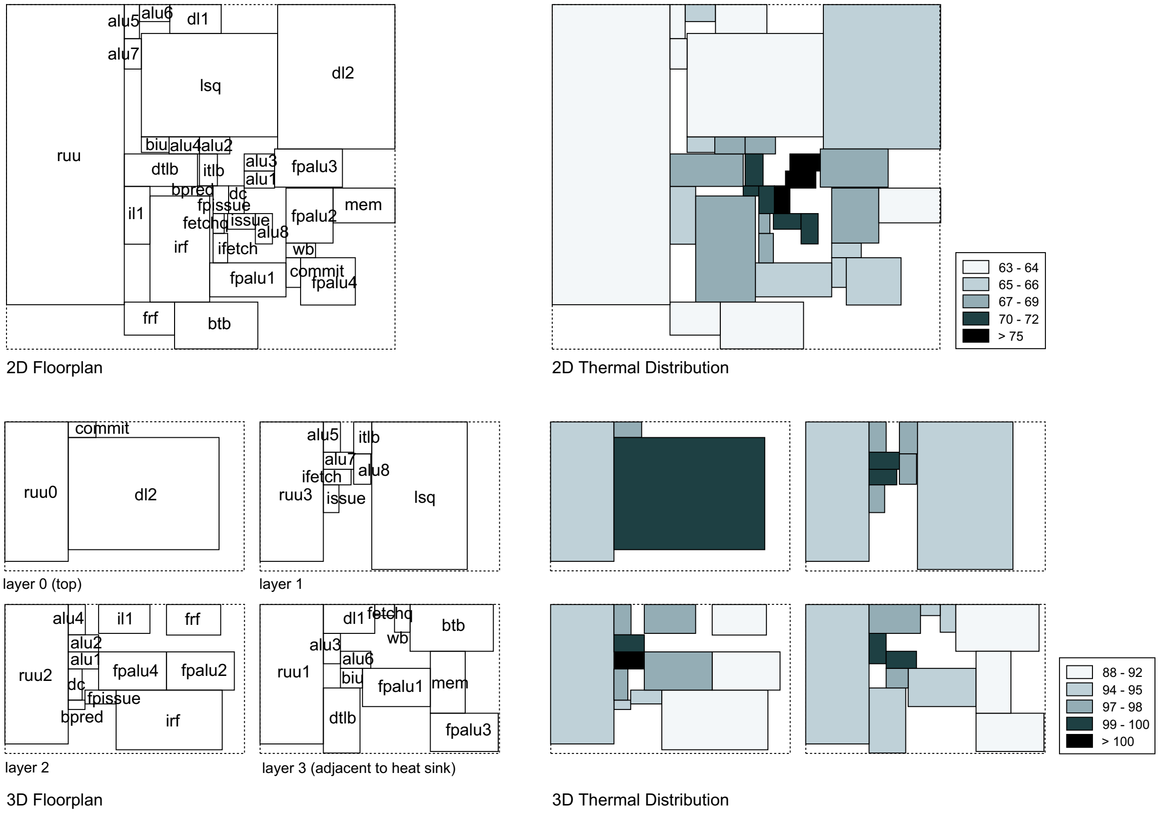


Figure 10: Snapshots of our 2D/3D floorplanning

for multi-objective, multi-layer floorplanning problem. Future work will try to address this problem more effectively. A possible solution is to utilize the whitespace for decoupling capacitors, thermal vias, buffers, etc. Our flow provides the users with the ability to modify the objective weights to suit their needs. This figure demonstrates that there is indeed thermal coupling between adjacent modules and that the thermal portion of the objective has attempted to separate the hottest modules while the performance portion of the objective has caused some of the hottest modules to remain grouped. This stays in line with the rapid dropoff in performance with decreased temperature shown in Figure 9.

Table 5 shows the top 10 microarchitectural modules under various metrics. Physical designers often are only able to view the modules at the floorplan level as little more than rectangles. Here we provide some more detailed information about each of the modules that make up the floorplan. This can provide better opportunities for optimization at the physical design level. The register update unit (RUU) [44] with a large number of read/write ports is larger in area than the next 2 largest modules combined, which is why it was split up for the multi-layer floorplans. The power density of the ALUs is higher than most of the other modules and hence their temperatures are also generally among the highest in the floorplan. The 3D floorplan is able to mitigate this by placing ALUs in different layers. Though several modules can have similar power consumption their temperatures may be different because their nearest neighbors can have a large impact on their final temperature. The leakage power profile among the modules is identical between the 2D and 3D floorplan except for the last two entries. This is because the logic styles of each module are more important in determining the relative leakage power than the variations in temperature. Table 6 shows the top 10 buses and interconnects under various metrics. It is interesting to note that the longest wire in the multi-layer floorplan is almost half as long as the longest wire in the single layer floorplan. The shortest wire list is dominated by inter-ALU connections. This is partly because the ALUs are generally small units and so the center to center distance for them is smaller but also because there are many data passing lines among the ALUs so they are very tightly connected.

In order to demonstrate the effect of various layer partitionings Table 7 is presented. As can be seen from the table bonding aware with area balance partitioning outperforms a pure area based approach on both IPC and temperature. It has slightly lower IPC than the bonding aware with profile weighted balance style partitioning but the area is completely unacceptable due to the fact that area balance is not considered. The wirelength and runtime of all approaches is comparable.

Table 3: Multi-objective floorplanning results with performance (P), maximum block temperature (T), area (A), wirelength (W), and runtime reported. The LP+SA-based floorplanner is used. Temperature is in $^\circ C$. Whitespace (WS) is reported as a percentage.

	2D floorplan							
	A+W		A+P		A+T		A+P+T	
bench	IPC	temp	IPC	temp	IPC	temp	IPC	temp
gzip	2.01	78.3	2.83	100.4	2.03	75.2	2.69	86.2
swim	0.52	64.3	0.85	78.4	0.54	63.0	0.66	70.5
vpr	0.95	87.6	1.19	113.8	0.82	82.3	1.15	95.9
art	0.38	67.9	0.62	83.3	0.39	65.4	0.51	74.4
mcf	0.07	63.0	0.09	76.9	0.07	62.1	0.10	69.4
equake	0.40	62.7	0.47	76.3	0.41	61.8	0.43	69.0
lucas	0.63	95.6	0.75	123.2	0.64	88.3	0.80	103.5
gap	1.17	70.1	1.24	87.8	1.18	68.1	1.32	77.3
bzip2	1.42	80.4	1.90	103.6	1.47	77.1	1.65	88.4
twolf	0.60	92.3	0.94	120.8	0.61	85.8	0.61	101.1
AVG	0.81	76.2	1.09	96.46	0.82	72.9	0.99	83.6
AREA (mm^2)	52.46		57.23		58.66		60.37	
WIRE (mm)	345.20		412.15		358.86		449.67	
TIME (sec)	174		188		1116		1064	
PIPE	22		19		27		23	
WS (%)	10		20		23		21	
	3D floorplan							
	A+W		A+P		A+T		A+P+T	
bench	IPC	temp	IPC	temp	IPC	temp	IPC	temp
gzip	2.74	104.7	3.98	125.9	2.75	98.9	2.85	104.7
swim	0.71	92.9	0.85	106.9	0.72	84.1	0.92	88.0
vpr	1.30	111.5	1.40	137.0	1.25	107.1	1.29	114.4
art	0.52	95.6	0.59	111.4	0.52	87.9	0.61	92.0
mcf	0.10	92.0	0.11	105.4	0.10	83.1	0.07	86.6
equake	0.54	91.7	0.58	105.0	0.55	82.6	0.67	86.2
lucas	0.87	116.9	0.92	145.3	0.88	113.0	1.19	123.0
gap	1.59	97.0	1.59	114.2	1.62	89.6	1.61	94.5
bzip2	1.94	106.8	2.05	129.0	1.98	101.5	2.33	107.4
twolf	0.81	114.6	1.03	142.2	0.84	111.0	1.02	118.9
AVG	1.11	102.4	1.31	122.2	1.12	95.8	1.26	101.6
AREA (mm^2)	22.20		23.63		25.45		26.45	
WIRE (mm)	217.20		323.43		252.08		247.02	
TIME (sec)	180		438		16913		20016	
PIPE	22		17		24		21	
WS (%)	9		16		25		23	

Table 4: Comparison among pure-SA, pure-LP, and LP+SA approaches. The objective used is a linear combination of performance, temperature, and area all with equal weight. Temperature is in $^{\circ}C$.

	2D floorplan					
	pure SA		pure LP		LP+SA	
bench	IPC	temp	IPC	temp	IPC	temp
gzip	2.38	102.2	1.94	80.19	2.69	86.2
swim	0.61	83.5	0.66	69.3	0.66	70.5
vpr	0.93	113.1	1.24	86.9	1.15	95.9
art	0.45	87.5	0.48	71.9	0.51	74.4
mcf	0.08	82.0	0.09	68.3	0.10	69.4
equake	0.47	81.6	0.49	68.1	0.43	69.0
lucas	0.75	122.6	0.79	93.8	0.80	103.5
gap	1.38	91.1	1.34	73.7	1.32	77.3
bzip2	1.68	105.2	1.59	81.8	1.65	88.4
twolf	0.70	118.6	0.68	90.1	0.61	101.1
AVG	0.94	98.7	0.93	78.4	0.99	83.6
AREA (mm^2)	60.90		314.72		60.37	
WIRE (mm)	388.13		524.81		449.67	
TIME (sec)	1225		826		1064	

	3D floorplan					
	pure SA		pure LP		LP+SA	
bench	IPC	temp	IPC	temp	IPC	temp
gzip	2.74	109.5	2.31	97.5	2.85	104.7
swim	0.71	91.8	0.70	86.7	0.92	88.0
vpr	1.07	119.8	1.24	103.4	1.29	114.4
art	0.52	95.7	0.51	89.0	0.61	92.0
mcf	0.10	90.4	0.10	85.9	0.07	86.6
equake	0.54	90.0	0.53	85.7	0.67	86.2
lucas	0.87	128.7	0.85	108.1	1.19	123.0
gap	1.59	98.9	1.49	90.9	1.61	94.5
bzip2	1.94	112.2	1.81	99.4	2.33	107.4
twolf	0.81	124.8	0.77	106.2	1.03	118.9
AVG	1.09	106.2	1.03	95.3	1.26	101.6
AREA (mm^2)	21.59		70.64		26.45	
WIRE (mm)	230.47		207.57		247.02	
TIME (sec)	25157		18207		20016	

Table 5: The top 10 list of blocks under various metrics.

					2D floorplan			
rank	area (mm^2)		power (mW/mm^2)		temperature ($^\circ C$)		leakage (mW)	
1	RUU	16.38	IALU1	15408	IALU1	83.5	L2 cache	0.9020
2	L2 cache	7.83	BPRED	1971	ITLB	78.6	ITLB	0.2470
3	LSQ	6.53	COMMIT	1930	L1 icache	76.3	DTLB	0.2470
4	IRF	2.94	FPISSUE	1930	FETCHQ	75.5	L1 icache	0.0588
5	BTB	1.81	ITLB	1049	FPALU1	75.4	L1 dcache	0.0588
6	FPALU 2	1.20	IALU2	1034	MEM	73.0	BTB	0.0088
7	FPALU 3	1.20	IALU3	884	COMMIT	72.5	FETCHQ	0.0035
8	FPALU 4	1.20	IALU4	746	IALU5	72.1	FPALU2	0.0014
9	DTLB	1.10	L1 cache	730	FPALU2	72.1	FPALU3	0.0014
10	MEM	1.00	IALU5	630	IALU7	70.8	FPALU1	0.0014
					3D floorplan			
rank					temperature ($^\circ C$)		leakage (mW)	
1					IALU1	104.7	L2 cache	0.9343
2					MEM	103.7	ITLB	0.2559
3					IALU5	103.1	DTLB	0.2559
4					ITLB	103.1	L1 icache	0.0609
5					L2 cache	102.2	L1 dcache	0.0609
6					IALU4	102.0	BTB	0.0091
7					FPALU4	101.7	FETCHQ	0.0036
8					IALU8	100.0	FPALU3	0.0015
9					IALU2	99.6	FPALU1	0.0015
10					IALU3	97.3	FPALU2	0.0015

Table 6: The top 10 list of wires under various metrics.

			2D floorplan			
rank	access frequency		wirelength (mm)		wirelength (mm)	
1	ITLB-FETCHQ	1.0	IRF-IALU5	8.575	IALU7-IALU6	0.53
2	IF-DC	1.0	IRF-IALU1	7.132	IF-DC	0.62
3	BTB-IF	1.0	DL1-RUU	6.944	DC-ISSUE	0.62
4	IL1-FETCHQ	1.0	FPALU3-RUU	6.710	IALU2-IALU3	0.65
5	FETCHQ-IF	1.0	RUU-FPALU3	6.710	IALU4-IALU8	0.65
6	DC-ISSUE	1.0	IRF-FPALU1	6.414	IALU4-IALU6	0.67
7	DL2-DL1	1.0	DL1-IALU5	6.414	ITLB-FETCHQ	0.96
8	WB-COMMIT	1.0	IRF-IALU7	5.797	IL1-FETCHQ	1.00
9	DTLB-RUU	1.0	IALU6-RUU	5.730	IALU4-IALU7	1.16
10	DL1-RUU	1.0	DL2-IL1	5.659	IALU6-IALU8	1.33
			3D floorplan			
rank			wirelength (mm)		wirelength (mm)	
1			IALU6-RUU	4.696	IALU1-FETCHQ	0.23
2			FPALU3-RUU	4.479	IALU5-IALU1	0.33
3			IRF-IALU6	3.962	IALU5-IALU2	0.35
4			WB-COMMIT	3.959	IALU8-IALU3	0.36
5			DTLB-RUU	3.688	IRF-FPALU1	0.57
6			DL1-RUU	3.613	IALU4-IALU1	0.65
7			IRF-IALU5	3.482	IALU8-IALU1	0.67
8			IRF-IALU2	3.462	IALU2-IALU1	0.67
9			RUU-FPALU1	3.423	IALU2-IALU4	0.67
10			DL2-IL1	3.395	IALU4-IALU5	0.69

Table 7: A comparison between the different partitioning styles. The hybrid A+P+T objective is used with combined LP+SA approach.

	Area Greedy		Bonding Aware w/ Profile Weight		Bonding Aware w/ Area	
bench	IPC	temp	IPC	temp	IPC	temp
gzip	2.98	108.9	2.88	108.8	2.85	104.7
swim	0.77	93.0	0.87	96.8	0.92	88.0
vpr	1.16	117.8	1.54	112.9	1.29	114.4
art	0.57	95.6	0.65	99.7	0.61	92.0
mcf	0.11	91.8	0.12	92.9	0.07	86.6
equake	0.59	89.8	0.66	95.3	0.67	86.2
lucas	0.96	127.2	1.06	117.1	1.19	123.0
gap	1.77	99.8	1.88	100.2	1.61	94.5
bzip2	2.14	110.4	2.29	109.2	2.33	107.4
twolf	0.90	126.9	0.95	118.0	1.03	118.9
AVG	1.20	106.1	1.29	105.1	1.26	101.6
AREA (mm^2)	22.68		52.54		26.45	
WIRE (mm)	270.73		263.26		247.02	
TIME (sec)	19872		20102		20016	

CHAPTER VII

CONCLUSIONS

In this thesis, the first multi-objective *microarchitecture-level* floorplanning algorithm for high-performance, high-reliability microprocessors targeting both 2D and 3D ICs was presented. Performance and thermal objectives were simultaneously considered such that the automated floorplanner provided a balanced or goal-directed processor organization that achieved user-specified design objectives. Moreover, leakage modeling was integrated into the thermal analyzer, which allowed monitoring of the temperature/leakage interaction to prevent thermal runaway. In addition, the modules were partitioned into multiple layers while considering the through-via requirements for face-to-face and face-to-back bonding styles. The hybrid approach that combines Linear Programming and Simulated Annealing that was proposed proved to be very effective in obtaining high-quality solutions in reasonable runtime.

REFERENCES

[1] Agarwal, V., Hrishikesh, M. S., Keckler, S. W., and Burger, D., “Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures,” in *Proc. IEEE Int. Conf. on Computer Architecture*, 2000.

[2] Austin, T. M., “Simplescalar tool suite.” http:/www.simplescalar.com, August 2005.

[3] Balakrishnan, K., Nanda, V., Easwar, S., and Lim, S. K., “Wire Congestion And Thermal Aware 3D Global Placement,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2005.

[4] Brooks, D., Tiwari, V., and Martonosi, M., “Wattch: A framework for architectural-level power analysis and optimizations,” in *Proc. IEEE Int. Conf. on Computer Architecture*, 2000.

[5] Brooks, D. and Martonosi, M., “Dynamic thermal management for high-performance microprocessors,” in *Proceedings of the Seventh International Symposium on High-Performance Computer Architecture*, (Monterrey, Mexico), p. 171, IEEE Computer Society, 2001.

[6] Casu, M. and Macchiarulo, L., “Floorplanning for throughput,” in *Proc. Int. Symp. on Physical Design*, 2004.

[7] Chen, G. and Sapatnekar, S., “Partition-driven standard cell thermal placement,” in *Proc. Int. Symp. on Physical Design*, 2003.

[8] Cheng, L., Deng, L., and Wong, M., “Floorplan Design for 3-D VLSI Design,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2005.

[9] Cheng, L., Hung, W., Yang, G., and Song, X., “Congestion Estimation for 3-D Circuit Architectures,” *IEEE Trans. On Circuits and Systems II: Express Briefs*, 2004.

[10] Chu, C. N. and Wong, D. F., “A matrix synthesis approach to thermal placement,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1998.

[11] Cong, J., Jagannathan, A., Reinman, G., and Romesis, M., “Microarchitecture evaluation with physical planning,” in *Proc. ACM Design Automation Conf.*, 2003.

[12] Cong, J., Wei, J., and Zhang, Y., “A Thermal-Driven Floorplanning Algorithm for 3D ICs,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2004.

[13] Cong, J. and Zhang, Y., “Thermal-Driven Multilevel Routing for 3-D ICs,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2005.

[14] Das, S., Chandrakasan, A., and Reif, R., “Design Tools for 3-D Integrated Circuits,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2003.

[15] Deng, Y. and Maly, W., "Physical design of the 2.5D stacked system," in *Proc. IEEE Int. Conf. on Computer Design*, 2003.

[16] Dropsho, S., Kursun, V., Albonesi, D., Dwarkadas, S., and Friedman, E., "Managing static leakage energy in microprocessor functional units," in *Proc. Annual Int. Symp. Microarchitecture*, 2004.

[17] Duarte, D., Tsai, Y., Vijaykrishnan, N., and Irwin, M., "Evaluating Run-Time Techniques for Leakage Power Reduction," in *Proc. Asia and South Pacific Design Automation Conf.*, 2002.

[18] Duarte, D., Vijaykrishnan, and Erwin, M. J., "A clock power model to evaluate the impact of architectural and technology optimizations," *IEEE Transactions on VLSI Systems, Volume 10, Issue 6*, pp. 844–855, Dec. 2002.

[19] Eble, J. C., De, V. K., Wills, D. S., and Meindl, J. D., "A Generic System Simulator (GENESYS) for ASIC Technology and Architecture Beyond 2001," in *Int'l ASIC Conference*, 1996.

[20] eCACTI. http://www.ics.uci.edu/∼maheshmn/eCACTI/main.htm, March 2005.

[21] Ekpanyapong, M., Minz, J., Watewai, T., Lee, H.-H., and Lim, S. K., "Profile-guided microarchitectural floorplanning for deep submicron processor design," in *Proc. ACM Design Automation Conf.*, 2004.

[22] Goplen, B. and Sapatnekar, S., "Thermal Via Placement in 3-D ICs," in *Proc. Int. Symp. on Physical Design*, 2005.

[23] Goplen, B. and Sapatnekar, S., "Efficient Thermal Placement of Standard Cells in 3D ICs using a Force Directed Approach," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, 2003.

[24] He, L., Liao, W., and Stan, M., "System Level Leakage Reduction Considering Leakage and Thermal Interdependency," in *Proc. ACM Design Automation Conf.*, 2004.

[25] Ho, R., Mai, K. W., and Horowitz, M. A., "The Future of Wires," *Proceedings of the IEEE*, 2001.

[26] Horn, S. B., "Vertically Integrated Sensor Arrays VISA," in *Defense and Security Symposium*, 2004.

[27] HotSpot. http://lava.cs.virginia.edu/HotSpot, February 2006.

[28] Huang, M., Renau, J., Yoo, S.-M., and Torrellas, J., "A framework for dynamic energy efficiency and temperature management," in *Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture*, (Monterey, California), pp. 202–213, 2000.

[29] Hung, W., Xie, Y., Vijaykrishnan, N., Addo-Quaye, C., Theocharides, T., and Irwin, M., "Thermal-aware floorplanning using genetic algorithms," in *Proc. Int. Symp. on Quality Electronic Design*, 2005.

[30] Kaxiras, S., Hu, Z., and Martonosi, M., "Cache decay: exploiting generational behavior to reduce cache leakage power," in *Proceedings of the 28th annual international symposium on Computer architecture*, (Gteborg, Sweden), pp. 240–251, 2001.

[31] Kaya, I., Olbrich, M., and Barke, E., "3-D Placement Considering Vertical Interconnects," in *Proc. IEEE Int. SOC Conf.*, 2003.

[32] Kim, N., Flautner, K., Blaauw, D., and Mudge, T., "Drowsy instruction caches: leakage power reduction using dynamic voltage scaling and cache sub-bank prediction," in *Proc. Annual Int. Symp. Microarchitecture*, 2002.

[33] Liao, W., Li, F., and He, L., "Microarchitecture level power and thermal simulation considering temperature," in *Proc. Int. Symp. on Low Power Electronics and Design*, 2003.

[34] Long, C., Simonson, L., Liao, W., and He, L., "Floorplanning optimization with trajectory piecewise-linear model for pipelined interconnects," in *Proc. ACM Design Automation Conf.*, 2004.

[35] Minz, J., Lim, S. K., and Koh, C. K., "3D Module Placement for Congestion and Power Noise Reduction," in *Proc. Great Lakes Symposum on VLSI*, 2005.

[36] Minz, J., Wong, E., and Lim, S. K., "Thermal and Power Integrity-aware Floorplanning for 3D Circuits," in *Proc. IEEE Int. SOC Conf.*, 2005.

[37] Murata, H., Fujiyoshi, K., Nakatake, S., and Kajitani, Y., "Rectangle packing based module placement," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 472–479, 1995.

[38] Nookala, V., Chen, Y., Lilja, D., and Sapatnekar, S., "Microarchitecture-Aware Floorplanning Using a Statistical Design of Experiments Approach," in *Proc. ACM Design Automation Conf.*, 2005.

[39] Obermeier, B. and Johannes, F., "Temperature-aware global placement," in *Proc. Asia and South Pacific Design Automation Conf.*, 2004.

[40] Pavlidis, V. and Friedman, E., "Interconnect Delay Minimization through Interlayer Via Placement in 3-D ICs," in *Proc. Great Lakes Symposum on VLSI*, 2005.

[41] Shivakumar, P. and Jouppi, N. P., "CACTI 3.0: An Integrated Cache Timing, Power, and Area Model," Tech. Rep. 2001.2, HP Western Research Labs, 2001.

[42] SIA, "National Techonology Roadmap for Semiconductors," 2003.

[43] Skadron, K., Stan, M., Huang, W., Velusamy, S., Sankaranarayanan, K., and Tarjan, D., "Temperature-aware microarchitecture," in *Proc. IEEE Int. Conf. on Computer Architecture*, pp. 2–13, 2003.

[44] Sohi, G. and Vajapeyam, S., "Instruction issue logic for high performance interruptable pipelined processors," *Proceedings of the 14th Annual International Symposium on Computer Architecture*, 1987.

[45] Tanprasert, T., "An Analytical 3-D Placement That Reserves Routing Space," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2000.

[46] Tsai, C. and Kang, S., "Cell-level placement for improving substrate thermal distribution," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2000.

[47] Tsai, Y., Ankadi, A., Vijaykrishnan, N., Irwin, M., and Theocharides, T., "ChipPower: An Architecture-Level Leakage Simulator," in *Proc. IEEE Int. SOC Conf.*, 2004.

[48] Umemoto, M., Tanida, K., Nemoto, Y., Hoshino, M., Kojima, K., Shirai, Y., and Takahashi, K., "High-Performance Vertical Interconnection for high-density 3D Chip Stacking Package," in *IEEE Electronic Components and Technology Conf.*, 2004.

[49] Zhang, R., Roy, K., Koh, C.-K., and Janes, D. B., "Exploring SOI device structures and interconnect architectures for 3-dimensional integration," in *Proc. ACM Design Automation Conf.*, 2001.