國立交通大學

電子工程學系 電子研究所

碩 士 論 文

應用於三維可程式邏輯閘陣列之
熱感知擺放與繞線演算法

Thermal-Aware Placement and Routing for 3D FPGAs

研 究 生：張瀚元

指導教授：黃俊達 博士

中 華 民 國 一００ 年 九 月

應用於三維可程式邏輯閘陣列之
熱感知擺放與繞線演算法

Thermal-Aware Placement and Routing for 3D FPGAs

研究生：張瀚元　　Student: Han-Yuan Chang

指導教授：黃俊達 博士　　Advisor: Dr. Juinn-Dar Huang

國立交通大學

電子工程學系　電子研究所

碩士論文

A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical & Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Electronics Engineering & Institute of Electronics

September 2011
Hsinchu, Taiwan, Republic of China

中華民國一００年九月

應用於三維可程式邏輯閘陣列之
熱感知擺放與繞線演算法

研究生：張瀚元　　指導教授：黃俊達 博士

國立交通大學
電子工程學系 電子研究所碩士班

摘　　要

新興的三維技術被認為是獲得更好的系統性能及更易於整合的解決方案，其堆疊多個晶粒(die)至單一晶片(chip)並利用直通矽穿孔(through-silicon vias, TSVs)做為垂直方向的連接。另一方面，可程式邏輯閘陣列(FPGAs)具有許多優點，是目前產品設計的主流選項之一。因此很自然的，三維可程式邏輯閘陣列(3D FPGAs)可以更進一步提升系統效能。然而，在三維整合技術裡，較高的功率密度(power density)與較長的散熱途徑(heat dissipation path)使得散熱問題較傳統的二維積體電路嚴重。因此發展具備熱感知(thermal-aware)的三維可程式邏輯閘陣列自動合成框架(framework)是相當重要的。針對這個目標，我們在這篇論文提出一系列適用於三維可程式邏輯閘陣列(3D FPGAs)的精準細微(*fine-grained*)熱電阻模型以及稱為 *TherWare* 的熱感知擺放(placement)與繞線(routing)演算法。在擺放時，我們不僅依照邏輯方塊(logic tile)之間的影響與每個方塊位置的散熱途徑來分配對應的邏輯區塊(Configurable Logic Block, CLB)，還會設法抑制因過長導線所增加的連線功率(interconnect power)；此外，在繞線階段更將同時考慮總消耗功率最簡化及功率分布之均勻度對於溫度的影響。由實驗結果可以證實，相較於現行已知的熱感知合成框架，經由 *TherWare* 所產生的合成結果在只需要增加些許電路延遲與程式執行時間的情況之下，最佳化過後的系統其最高溫、溫度標準差及最大溫度梯度都能夠被大幅地改善。

Thermal-Aware Placement and Routing for 3D FPGAs

Student: Han-Yuan Chang Advisor: Dr. Juinn-Dar Huang

Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University

Abstract

The emerging 3D technology, which stacks multiple dies within a single chip and utilizes through-silicon vias (TSVs) as vertical connections, is considered a promising solution for achieving better performance and easy integration. Meanwhile, field programmable gate array (FPGA) is one of the mainstream design solutions with lots of advantages. Therefore, 3D FPGA is a natural extension for further performance optimization. However, in 3D integration technology, the thermal issue is exacerbated mainly due to larger power density and longer heat dissipation path. As a result, the thermal-aware framework has been getting lots of attention in electronic designs. For this purpose, we propose a set of precise *fine-grained* thermal resistive models and a thermal-aware backend (placement and routing) flow named *TherWare* dedicated to 3D FPGAs in this thesis. In the placement stage, we not only consider the power distribution of logic tiles and heat dissipation path for each tile but also prevent the increase of interconnect power due to longer wirelength. In the routing stage, both power minimization and power distribution are considered. Finally, the experimental results show that our proposed *TherWare* can significantly improve maximum temperature, temperature deviation and maximum temperature gradient only with a minor increase in delay and runtime compared with the prior arts.

Content

List of Tables

List of Figures

Chapter 1
Introduction

1.1 3D Integrated Circuits

Ever shrinking feature size and higher transistor density make chip designs larger and more complicated. However, beyond nano-scale CMOS technology, further device shrinking is getting more difficult due to physical limitations. Furthermore, the interconnect delay has become domination of the system performance on chip, as shown in Figure 1, the delay of global interconnects is much larger than that of gates at 32nm process.

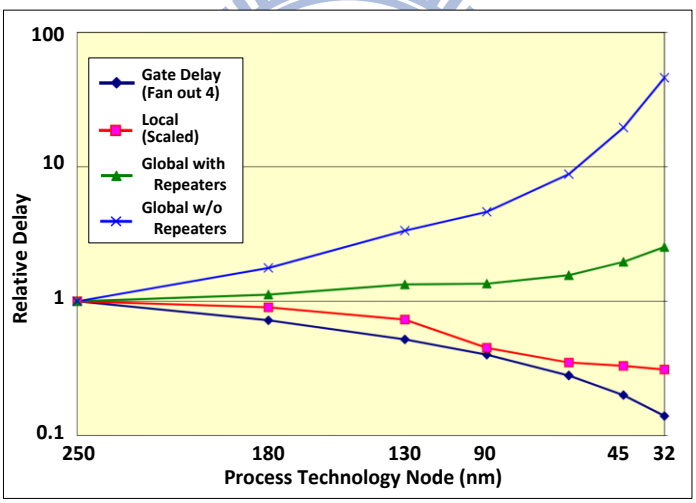


Figure 1. Relative delay vs. feature size [1].

Consequently, an alternative integration technology based on three-dimensional (3D) chip stacking emerges. This technology combines multiple chips through wafer/die bonding techniques [2][3]. By stacking chips and communicating inter-layer signals vertically, the length of global interconnect is significantly reduced and thus the performance improves after 3D integration as shown in Figure 2.

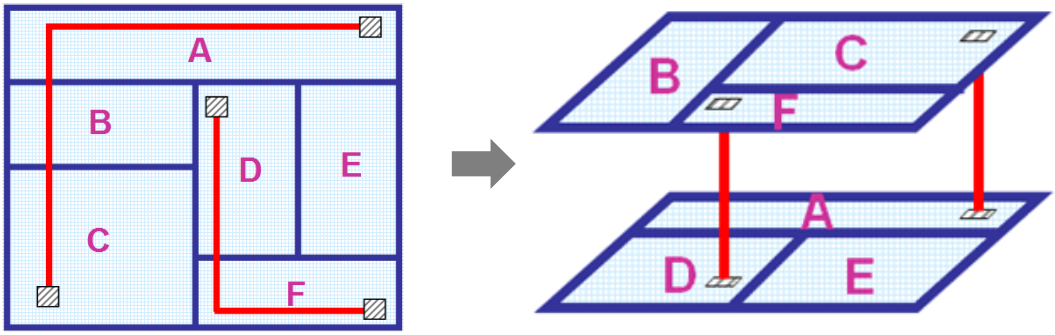


Figure 2. Global interconnects before and after 3D integration.

In order to combine multiple chips, there are two methods to accomplish communication links between different layers in vertical direction, namely, wire bonding and through-silicon vias (TSVs) technology, which have been discussed in recent years [2]–[6].

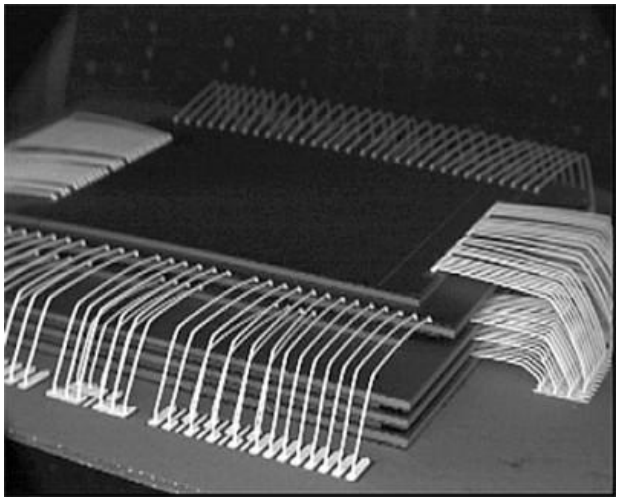


Figure 3. Wire bonding technology [7].

As shown Figure 3, wire bonding technology for a system-in-package (SiP) is a commonplace technique. However, the locations for wire-bonding are restricted on the periphery of a chip layer and the package substrate. Thus these kinds of 3D techniques are facing the problems such as limited number of pins for vertical connections, long and slow vertical signal paths, and chip-package co-designs.

Meanwhile, the through-silicon via (TSV) technology is also considered a promising solution for 3D integration as shown in Figure 4. The TSV-based 3D ICs stack multiple dies on a single chip and use inter-die vias for vertical connections.

These vias can be located almost everywhere within a chip. Though the benefits offered by TSVs are extremely attractive, such as shorter global interconnects [8]–[11], lower interconnect power [12], smaller footprint [13] and better heterogeneous integration [2], there are still many challenges of TSV-based 3D integration, which include reliability, yield [13], power density, and above all, the huge area cost.

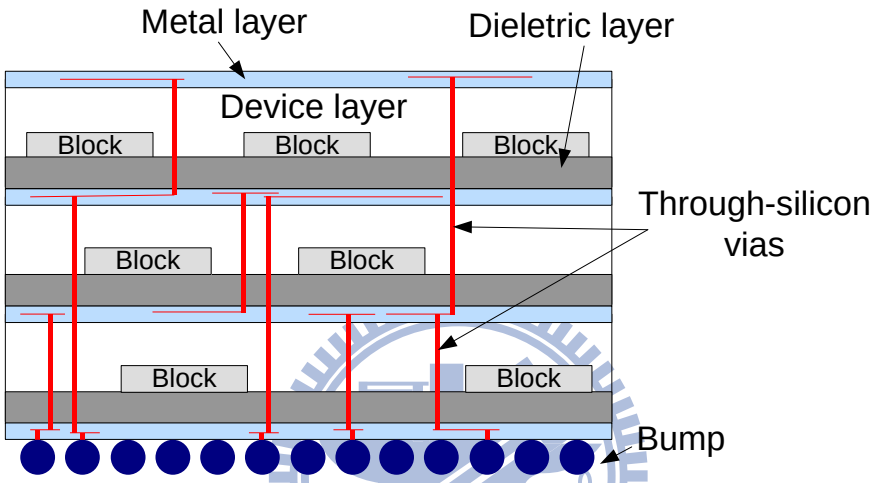


Figure 4. Through-silicon vias (TSVs) technology.

Nowadays, field programmable gate arrays (FPGAs) are still one of the mainstream design solutions with the advantage of shorter design/turnaround cycle, faster time-to-market, and lower non-recurring engineering cost. Unfortunately, compared with application-specific integrated circuits (ASICs), FPGAs generally require longer clock cycle time and more routing-resources for providing field reprogrammable capability. However, since 3D integration technology provides several unique advantages compared with the conventional 2D one, which accordingly makes 3D integration itself a promising solution to further advanced FPGA fabrics.

1.2 Field Programmable Gate Array

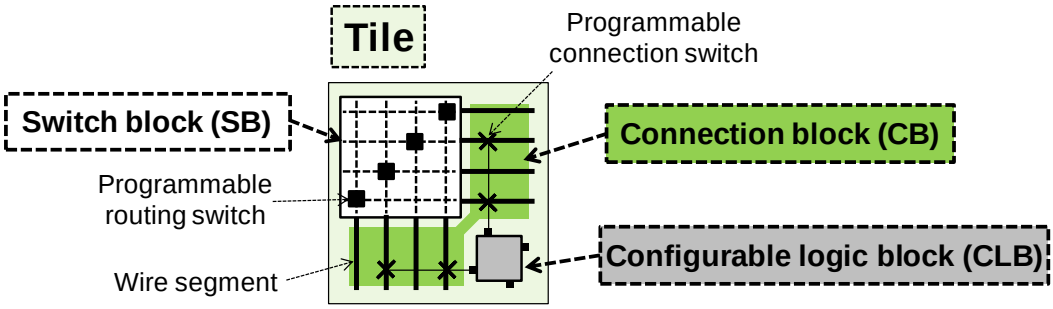


Figure 5. Tile structure.

A typical 2D regular FPGA architecture consists of a uniform 2D tile array. As shown in Figure 5, each tile contains i) a configurable logic block (CLB) – the basic logic functional unit, ii) a connection block (CB) – a set of programmable connection switches used to connect signals between CLBs and wires, and iii) a switch block (SB) – to bridge signals between different wires by programmable routing switches.

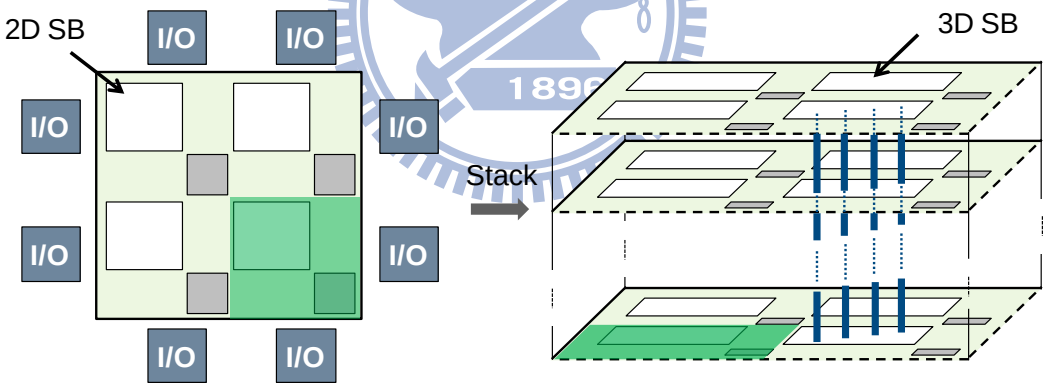


Figure 6. 2D and 3D FPGA architecture.

Meanwhile, the target regular 3D FPGA architecture in our work is depicted in Figure 6, which is basically a direct extension of its 2D counterpart. As in [14], a 3D FPGA consists of several identical 2D FPGA layers (i.e., same CLB/CB structure) stacking vertically and enhanced 3D SBs. That is, SBs are extended from 2D to 3D with extra Z-dimensional connectivity for inter-layer signal communication through TSVs.

1.3 Thermal Challenge

As shown in Figure 7, in 3D integration technology, the thermal issue is exacerbated mainly for two reasons – i) vertically overlapping blocks lead to a large increase of power density [15][16]; ii) except for the top layer adjacent to a heat sink, stacking multiple chips vertically results in a longer heat dissipation path. High temperature inside a chip degrades carrier mobility, increases metal resistivity, and lifts leakage power. In other words, higher chip temperature declines system performance, consumes more energy, and even lowers design reliability. As a result, the thermal-aware framework has been getting lots of attention in electronic designs, especially for 3D ICs.

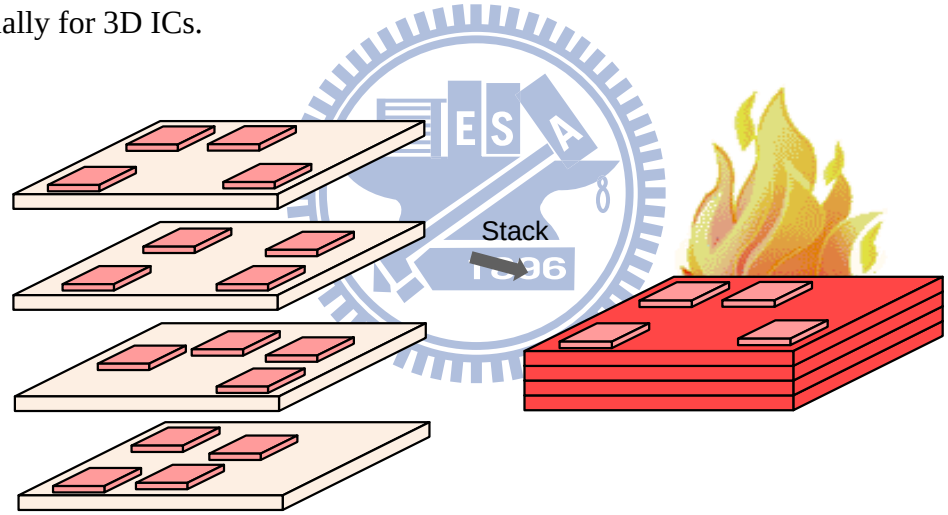


Figure 7. Thermal issue for 3D ICs.

1.4 Contribution

In this thesis, we first present that the power optimization is not sufficient for temperature optimization during placement and routing process. Then, we point out that even though the heat is primarily dissipated through vertical path in 3D ICs, the lateral heat flow cannot be neglected. It inspires us to develop a thermal-aware placement and routing algorithm, named TherWare. Both placement and routing

algorithm concentrate to minimize maximum temperature, temperature deviation and maximum temperature gradient, while keeping the delay and runtime overhead within few percent. TherWare placement is based on simulated annealing algorithm; three guidelines are integrated in thermal cost – distributing power uniformly, finding better position for potentially hotter tiles, as well as preventing excessive increase of interconnect power. TherWare routing is based on Pathfinder negotiated congestion algorithm [27], which takes the power overhead and power distribution into consideration.

Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, EDA flow and problem formulation are represented first, then a set of fine-grained thermal resistive models with different granularities, named *FG-8*, *FG-4* and *FG-2*, respectively, are proposed. We introduce three related works along with the motivation, and propose our TherWare placement and routing in Chapter 3 and Chapter4. In Chapter 5, the experimental environment, two experimental results and case study are represented. Finally, the concluding remarks are given in Chapter 6.

Chapter 2
Preliminaries

In this chapter, before we present the thermal-aware EDA flow for thermal-aware framework, the typical EDA flow should be introduced first. Then, we describe the problem formulation for this thesis.

2.1 EDA Flow

2.1.1 Typical EDA Flow

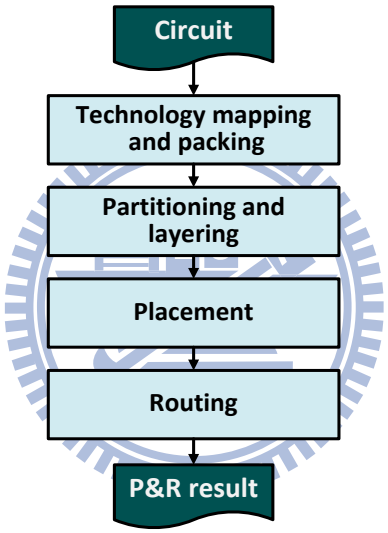


Figure 8. Typical EDA flow.

Figure 8 shows the typical EDA flow [14][17]. The first stage is technology mapping and packing; LUTs and registers are packed into basic logic elements (BLEs), and then the multiple BLEs are clustered into a netlist of CLB. In the second stage, these CLBs are divided into several partitions; each partition is assigned to different layers for minimized number of TSVs. Third stage places the CLBs to available hardware, the goal of this stage is to minimize the wirelength and delay. Final stage, determines which routing-resource should be used to connect all the CLB input and output pins required by the circuit based on placement result. Nevertheless,

after thermal analysis, the placement and routing result obtained from this flow contains many hotspots because thermal issue did not considered while synthesizing.

2.1.2 Thermal-Aware EDA Flow

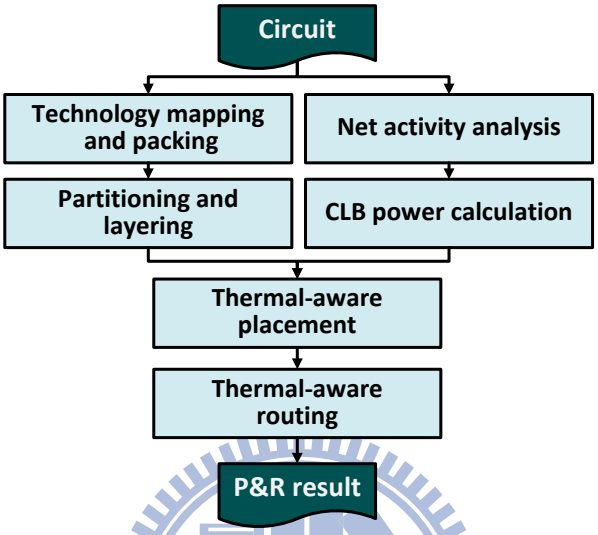


Figure 9. Thermal-aware EDA flow.

Figure 9 shows the thermal-aware EDA flow. The difference between thermal-aware EDA flow and typical EDA flow is that thermal-aware EDA flow needs two additional information – i) the switching activity of nets; ii) the power consumption of CLBs. By the information, thermal-aware placer and router can realize that which tile has higher probability to generate hotspots because it is crossed by the nets with higher switching activity or placed by a CLB with higher power consumption. Then, thermal-aware placer and router will place the CLBs and route the nets in a way that minimizes the temperature, such as maximum temperature, temperature deviation and maximum temperature gradient. As a result, after evaluating thermal information, we can obtain a better temperature profile.

2.2 Problem Formulation

Given a netlist of CLBs, 3D FPGA architecture, switching activity of nets and power consumption of CLBs, our goal is to find a placement result and a routing result under the constraints that – i) each logic hardware is placed by one CLB at most; ii) each net occupies uniquely a routing-resource. The most important objective of this thesis is that maximum temperature, temperature deviation and maximum temperature gradient are minimized with acceptable delay overhead.

2.3 Thermal Model

2.3.1 Thermal Modeling of 3D IC

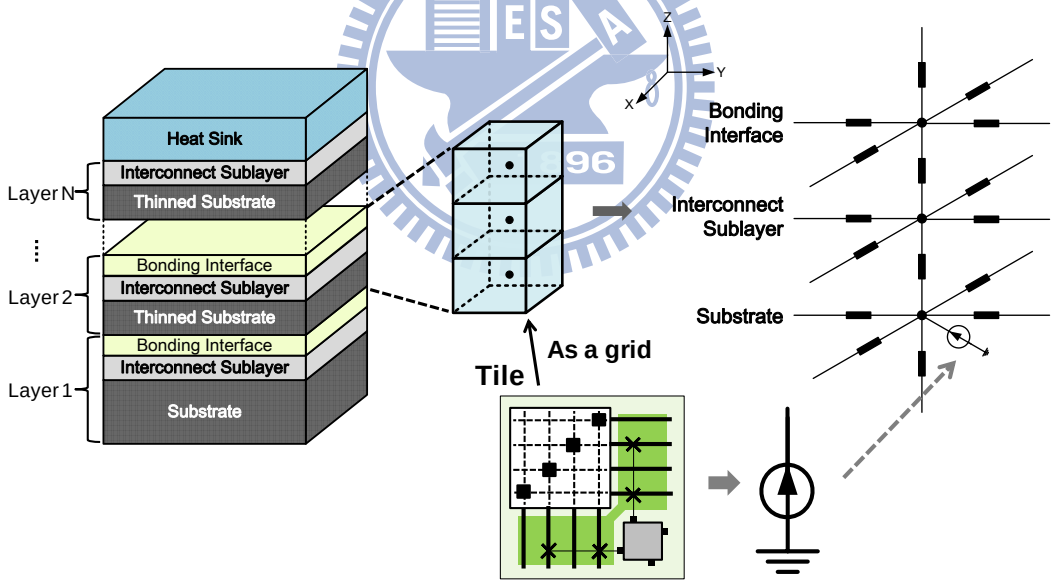


Figure 10. Thermal resistive model.

Figure 10 illustrates a typical 3D IC stacking configuration. Based on the thermal resistive model in [18], a single die is composed of three sublayers – the substrate where active devices reside, the interconnect sublayer where metal wires and vias reside, and the bonding interface attaching between two adjacent silicon dies. Heat generated from active devices is carried from substrates to a heat sink, and then

dissipated into the ambient air (25℃). Each die in a 3D design is first partitioned into a number of small regular grids with their own power densities. Each grid is further divided into several nodes according to the number of sublayers. Then a thermal resistance, whose value is determined based on both grid size and thermal properties of material, is attached between two adjacent nodes. Finally, the thermal resistive model applies thermal-electrical duality (as shown in Table 1) to generate a steady-state temperature profile for the given design.

Since our application targets 3D FPGAs, it is very natural to regard a tile as a grid due to regularity of FPGA. Therefore, the power consumption of each tile contains logic power and interconnect power. Logic power of each tile is only contributed by a placed CLB. Interconnect power of each tile is contributed by three elements – wire segment, SB and CB; if a wire segment in a tile is occupied by a net, these three elements will consume power because of hardware architecture.

Table 1. Thermal-electrical duality [19].

Thermal quantity	Unit	Electrical quantity	Unit
T, Temperature difference	K	*V*, Voltage difference	V
P, Power density	W	*I*, Current source	A
R_{th}, Thermal resistance	K/W	*R*, Electrical resistance	Ω

2.3.2 Proposed Fine-Grained Thermal Model

In this thesis, we set that the target 3D FPGAs are implemented in 45nm technology; the horizontal channel width is 32, and each vertical channel also contains 32 TSVs [20]; the pitch of each TSV is 6 μm [1]. Then by [17], we can estimate that the area of each tile is 47.8×47.8 μm^2. Other related parameters (e.g., thickness of each sublayer, thermal conductivity of each material, and so on) are properly set based on [21][22].

However, as stepping into the 3D era, the presence of TSVs makes accurate thermal modeling of 3D FPGA a bit complicated since thermal properties of all sublayers in a die would be substantially changed, and based on our settings that were introduced earlier, we can estimate that the ratio of area in TSVs is 50.4%. Therefore, it is assumed that TSVs account for nearly half area of a single tile and are uniformly distributed within a tile. As a consequence, we decide to further partition a tile into an array of even fine-grained grids and then construct a set of *fine-grained* thermal models. As shown in Figure 11, in order to examine the effects of models with different grid granularity, we divide a tile into a 2×2, 4×4, and 8×8 grid array, and name the model *FG-8*, *FG-4*, and *FG-8*, respectively.

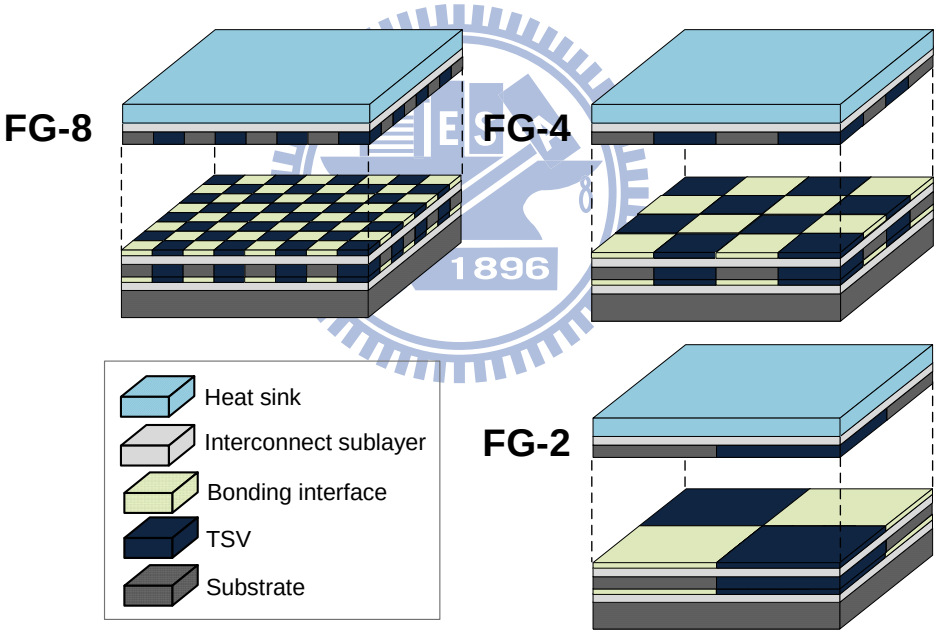


Figure 11. Proposed fine-grained thermal model.

2.3.3 Comparisons

For demonstrating the accuracy level of proposed fine-grained thermal models with different granularities, FG-8, FG-4, and FG-2, first we perform random logic mapping onto FPGA from one layer to eight layers (i.e., Z-dimension = 1~8); each FPGA contains 1000 tiles no matter how many layers it has (by properly setting

x/y-dimension); the logic utilization is set to 75%; the power of each mapped tile is set to the product of the power density of 2×10^6 W/m^2 [21]. After a thermal resistive network is built, hspice is then invoked to get a corresponding temperature profile. Every reported value in the following is an average of 5 random logic mapping runs.

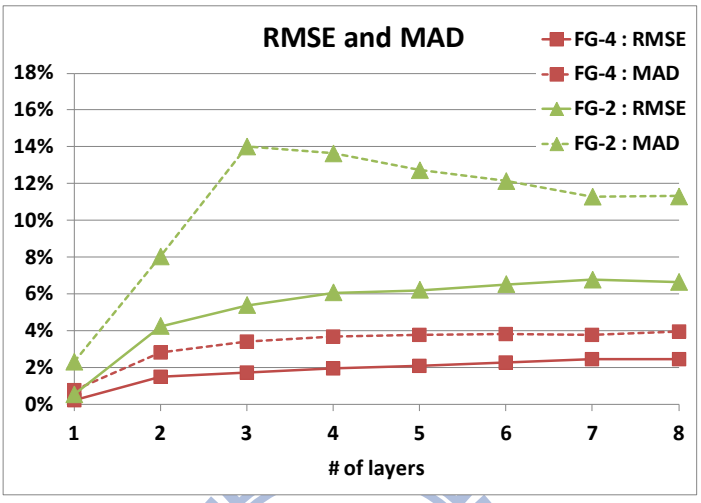


Figure 12. Node-to-node RMSE and MAD against FG-8.

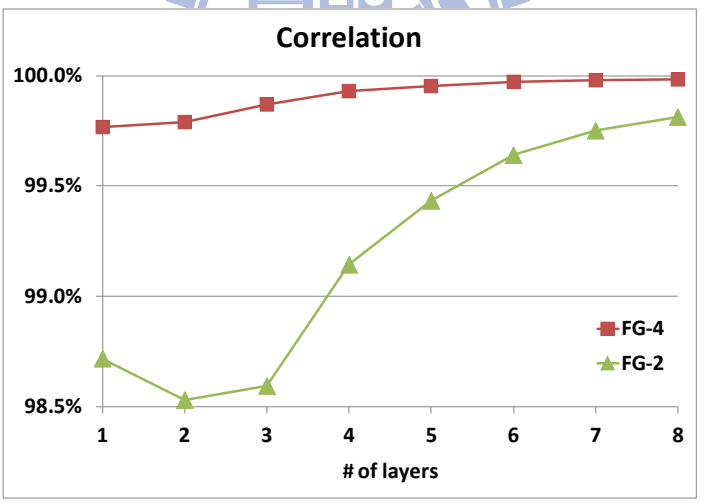


Figure 13. Node-to-node correlation against FG-8.

Figure 12 and Figure 13 report root mean square error (RMSE), maximum absolute difference (MAD) and correlation to show node-to-node differences with respect to FG-8. From these figures, FG-4 is just getting slightly inaccurate as the number of layers increases; however, compared with FG-4, FG-2 is more inaccurate than FG-4. In FG-4/FG-2, the root mean square error is less than 2.5%/6.7%, the maximum absolute difference is less than 3.9%/14.0%, and the correlation is more

than 99.8%/98.5%. Moreover, FG-8 takes 144.6 seconds to produce a temperature profile on average, while FG-4/FG-2 merely requires 19.9/17.0 seconds, which suggests a 7.3/8.5 times speedup against FG-8.

From above results, it is concluded that FG-4 can achieve a large speedup in runtime but with a tiny loss in accuracy. Therefore, FG-4 would be used as our thermal model.

Chapter 3
Placement Algorithm

3.1 Related Works

3.1.1 TPR

In 3D FPGAs, the first backend tool is three dimensional place and route (TPR) [14], which can supports timing-driven placement and routing. In TPR, placement approach is based on simulated annealing; logic blocks are selected and swapped or moved randomly during the placement until maximum number of iterations is reached. A cost function is used to evaluate the quality of placement result as shown in Equation (1).

$$Cost = \alpha \times Cost_{Wire} + \beta \times Cost_{Delay} \quad (1)$$

In Equation (1), it tries to minimize wirelength and delay, these two costs are calculated based on a timing analyzer and a net semi-perimeter metric wire length estimator. In addition, the factor α and β are used to trade-off between wirelength and delay.

3.1.2 3D MEANDER

3D MEANDER [23] is another design framework for 3D FPGAs, and it supports thermal-aware placement and routing. Since 3D MEANDER takes thermal issue into consideration, so the Equation (1) is modified to Equation (2), and the thermal cost in Equation (2) is presented in Equation (3).

$$Cost = \alpha \times Cost_{Wire} + \beta \times Cost_{Delay} + \gamma \times Cost_{Thermal} \quad (2)$$

$$Cost_{Thermal} = \sum_{i \in Net} Activity(i) \times Cost_{Wire}(i) \quad (3)$$

The concept of 3D MEANDER placement algorithm is to minimize interconnect dynamic power because heat source is transited from power consumption; that is, lower power may result in lower temperature. From Equation (3), the thermal cost is sum of multiplying switching activity by wirelength for each net; thus, the wirelength of nets with higher switching activity will be shorter by placing the CLBs connected through by this net to each other , leading to lower power consumption.

3.1.3 Z-tile

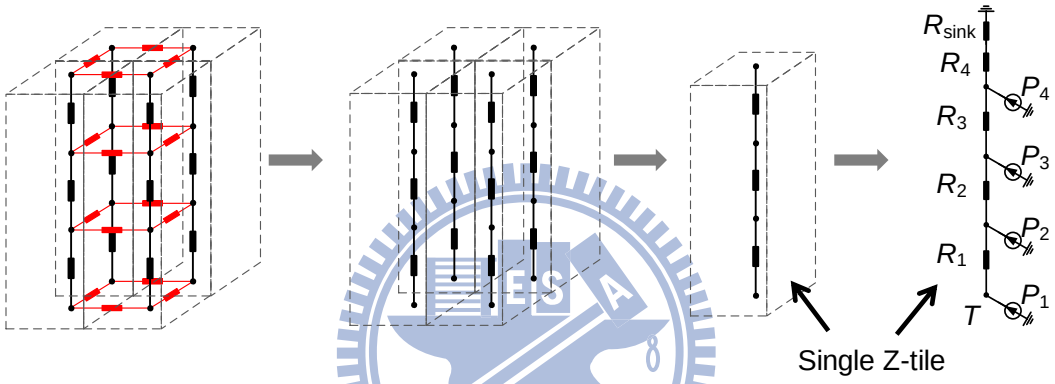


Figure 14. Construction of Z-tile model.

In the past decade, the Z-tile model is considered one of the most broadly used simplified thermal models, as depicted in Figure 14. Authors in [24] have observed that heat is primarily dissipated through vertical path in 3D ICs; therefore, for thermal model simplification, all lateral heat flows are intentionally ignored. It is also the reason why this simplified model is named the Z-tile model. By omitting all lateral thermal resistances, the Z-tile model facilitates fast temperature evaluation by Equation (4).

$$T = \sum_{i=1}^{4}(P_i \sum_{j=1}^{i} R_j) + R_{sink} \sum_{i=1}^{4} P_i \tag{4}$$

$$Cost_{Thermal} = T_{max} \tag{5}$$

For example, there are four single Z-tiles as shown in Figure 14, by calculating the temperature of each single Z-tile, the highest temperature value of all single Z-tiles will be put into the thermal cost (Equation (5)) for maximum temperature minimization, and thus has been widely used in many thermal-aware 3D ASIC design flows by put this thermal cost into Equation (2).

3.2 Motivation

In 3D MEANDER placement, for minimizing interconnect power, the wirelength of nets with higher switching activity must be shorter as Figure 15 shows; however, the CLBs which are connected through by these net, have higher power consumption generally. It causes that some regions will generate hotspots because these regions are placed by the CLBs with higher power consumption as shown in Figure 16. As a result, the maximum temperature does not decrease obviously even though the interconnect power is minimized.

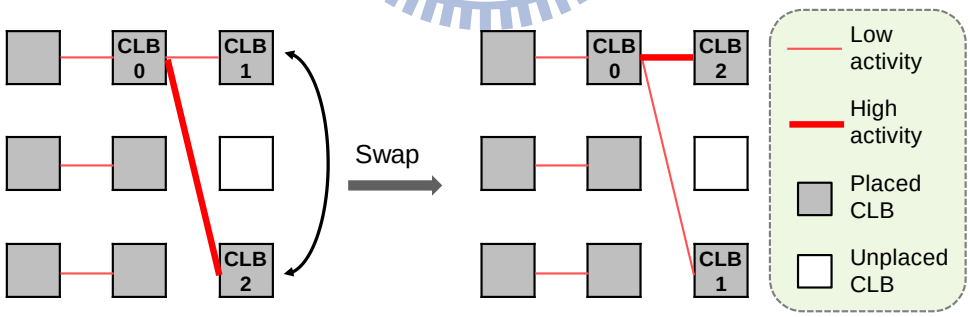


Figure 15. Interconnect power minimization.

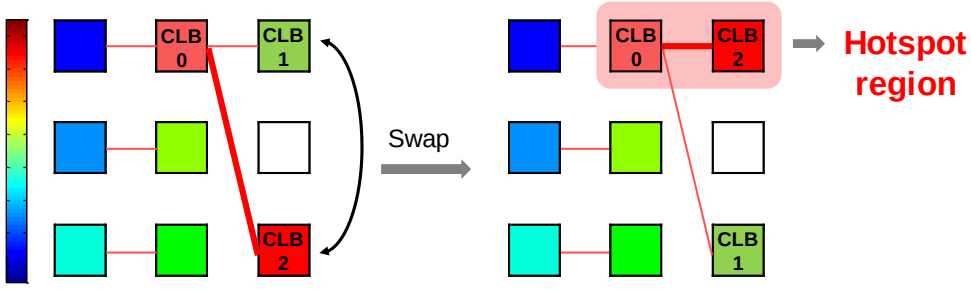


Figure 16. Effect of Interconnect power minimization.

For proving the drawback introduced earlier, we evaluate 3D MEANDER at 4-layer design for 20 largest MCNC benchmarks, and the logic utilization is set to 75%. Figure 17 shows the comparison between maximum temperature with total power of 3D MEANDER, the two curves are normalized to timing-driven – TPR, and we can observe that the some cases have very higher improvement of total power, but improvement of maximum temperature have not, such as diffeq, frisc, spla and pdc. Furthermore, we measure dependence (correlation) between these two curves, it is -0.22 which is a bit negative relationship. For these reasons, we think that the temperature optimization should focus on power distribution.

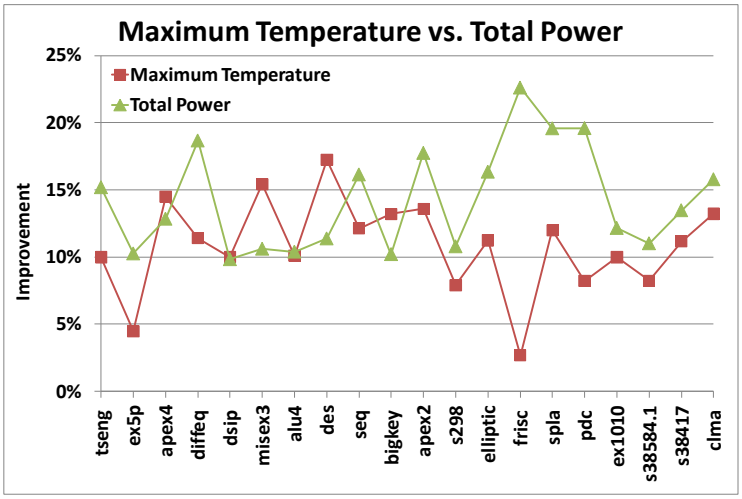


Figure 17. Comparison between max. temp with total power of 3D MEANDER.

In order to observe the effects of different power distributions on temperature, we perform two contrastive tile mapping patterns as shown in Figure 18, vertically *staggered* and *aligned*. For each mapping pattern, a set of configurations with different number of tiles in a big block are further considered. A configuration with a *group factor n* suggests there are $n \times n$ tiles within a block. In all experiments, the dimension of target 3D FPGA is fixed to 36×36×6, the utilization is set to 50%, all mapped tiles consume same power, and FG-4 thermal model is used for temperature evaluation. Notice that the vertically staggered can present the vertical heat flow is considered, and smaller group factor can presents the lateral heat flows is considered.

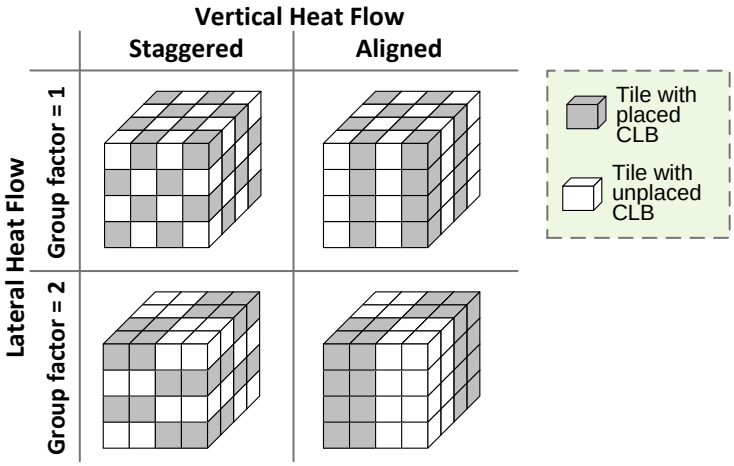


Figure 18. Staggered and aligned mapping patterns.

Figure 19 reports the results as a function of group factor. It is observed that maximum temperature is virtually independent of group factor for configurations using the staggered pattern. On the contrary, maximum temperature is significantly increased as group factor grows for configurations using the aligned pattern. Therefore, it seems practical that the Z-tile model only considers vertical heat flow and ignores all lateral heat flow.

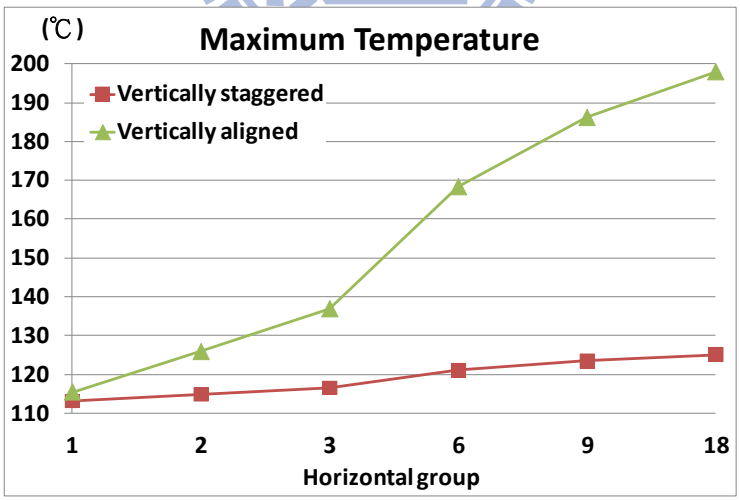


Figure 19. Maximum temperature with different group factors.

Nevertheless, as logic utilization increases, vertical heat flow cannot be staggered at some single Z-tiles as shown in Figure 20; that is, maximum temperature maybe significantly increased due to ignores all lateral heat flow. In other words, group factor cannot be controlled if we use Z-tile model as thermal cost during

placement. Therefore, we think that the lateral heat flow should not be neglected.

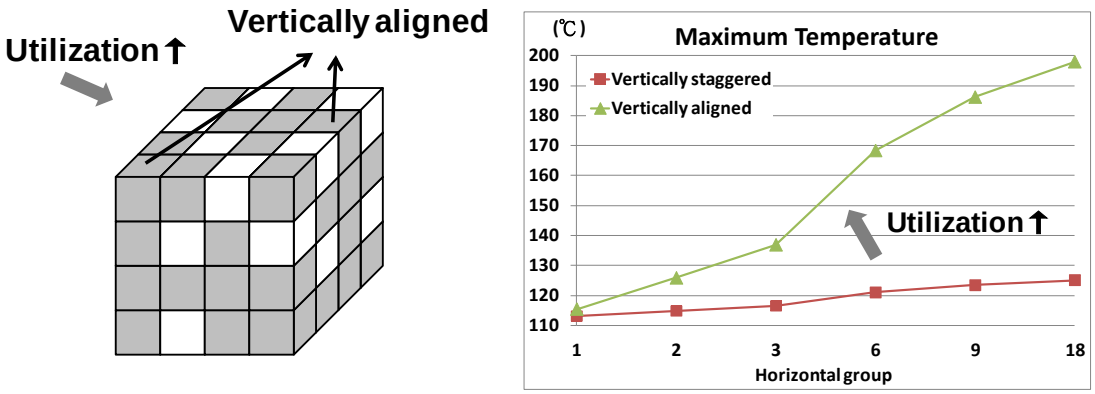


Figure 20. Effect of vertically aligned as utilization increases.

3.3 Proposed Algorithm – TherWare

In this section, we introduce our proposed thermal-aware placement algorithm – *TherWare*, which is based on simulated annealing, and the cost function is shown in Equation (2). The thermal cost of our TherWare placement is shown in Equation (6), and it regards for three guidelines – power uniformity ($Cost_{PU}$), heat dissipativity ($Cost_{HD}$) and interconnect power ($Cost_{IP}$), respectively.

$$Cost_{Thermal} = \gamma_1 \times Cost_{PU} + \gamma_2 \times Cost_{HD} + \gamma_3 \times Cost_{IP} \quad (6)$$

3.3.1 Power Uniformity

In this cost function, we want to keep power uniformity between several tiles with placed CLB. Generally, in timing-driven placement, the CLBs are placed to adjacent available hardware in order to shorten wirelength and delay. However, such a placement result has higher maximum temperature on temperature profile because of heat congestion. Therefore, the tiles with placed CLB should be uniformly spread out the entire FPGA as shown in Figure 21. In brief, for each tile with placed CLB, we want to minimize the power consumption of adjacent tiles.

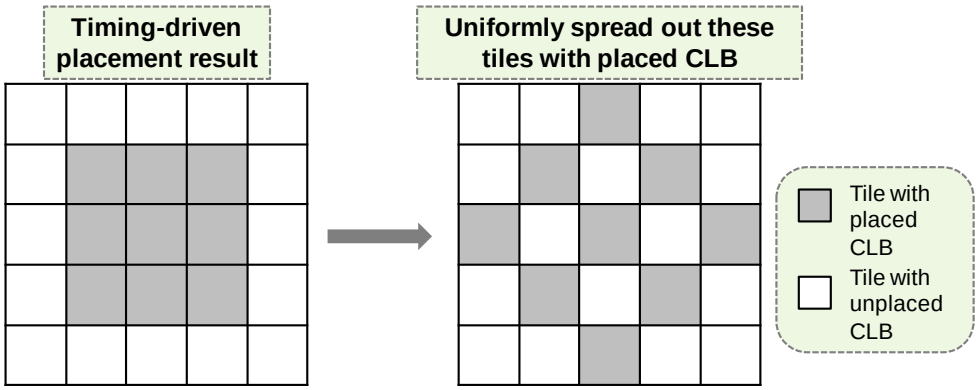
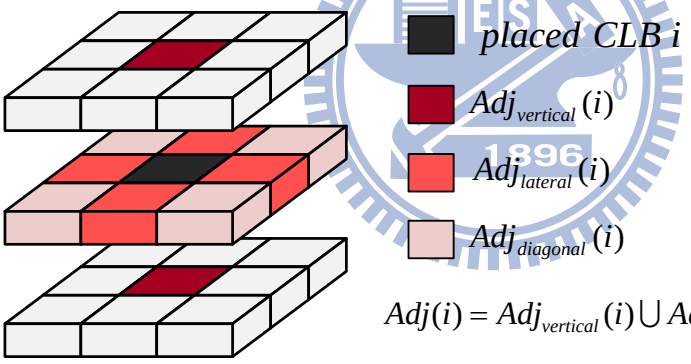


Figure 21. Uniform power distribution.

As shown in Figure 22, for a *placed CLB i*, the *Adj*(*i*) represents the set of its adjacent placed CLBs, and we classify this set into three subsets – i) $Adj_{vertical}(i)$ represents the set of vertical adjacent CLBs of *placed CLB i*, ii) $Adj_{lateral}(i)$ represents the set of lateral adjacent CLBs of *placed CLB i*, iii) $Adj_{diagonal}(i)$ represents the set of diagonal adjacent CLBs of *placed CLB i*.



$$Adj(i) = Adj_{vertical}(i) \cup Adj_{lateral}(i) \cup Adj_{diagonal}(i)$$

Figure 22. Definition of *Adj*(*i*).

In Equation (7) and Equation (8), the power uniformity cost is sum of adjacent power for all tiles with placed CLB; moreover, since vertical dissipating path is more important than lateral dissipating path, so we set the *Weight function* as shown in Equation (9), according to how important for dissipating heat.

$$Cost_{PU} = \sum_{i \in placed\ CLB} Adjacent\_Power(i) \tag{7}$$

$$Adjacent\_Power(i) = \sum_{j \in Adj(i)} Power(j) \times Weight(i, j) \tag{8}$$

$$Weight(i,j)=\begin{cases}1, & \text{CLB } j \in Adj_{lateral}\\ 0.7, & \text{CLB } j \in Adj_{diagonal}\\ 1.4, & \text{CLB } j \in Adj_{vertical}\end{cases} \tag{9}$$

3.3.2 Heat Dissipativity

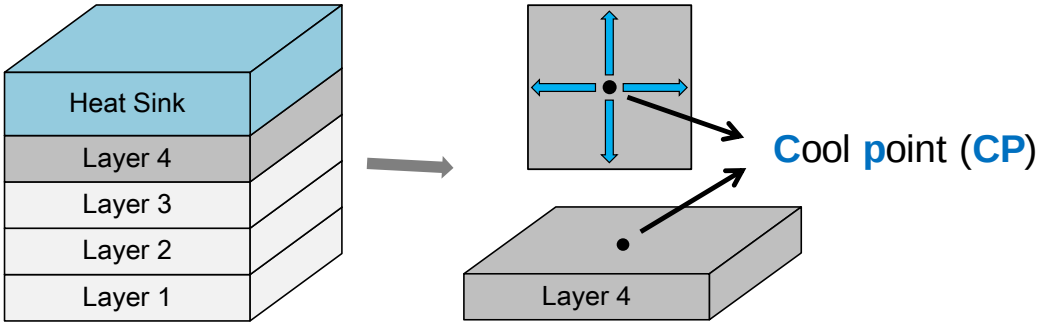


Figure 23. The cool point in 4-layer 3D IC.

In order to let the heat of potentially hotter tiles can be dissipated easily, the first step is to find out where are the position with the best heat dissipativity in 3D ICs. As shown in Figure 23, top-most layer is the closest to the heat sink, so it has better heat dissipativity than other layers, and the center of top-most layer has more area for dissipating heat; that is, center of top-most layer has the best heat dissipativity in entire 3D IC, where is named this position *cool point* (*CP*). Next, we want to estimate the potentially of heat for each tile with placed CLB. In section 2.3.1, we introduced that a tile contains logic power and interconnect power, so the pins activity must be considered because the pins of placed CLB are terminal of some nets, and these nets probably consume high interconnect power in this tile after routing, and higher interconnect power represents that this tile has higher potential of heat; for example, the tile 2 has higher potential of heat than tile 1 as shown in Figure 24.

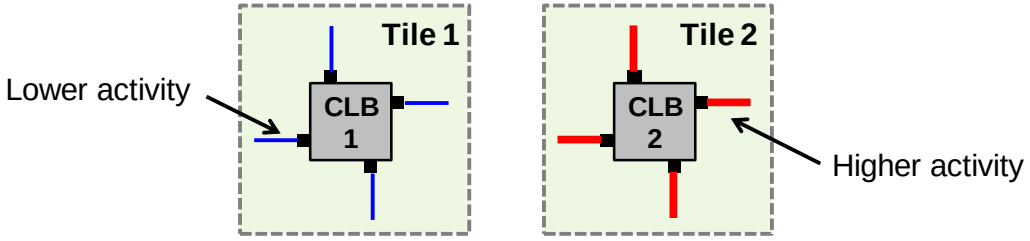


Figure 24. Comparison of two tiles.

As shown in Equation (10) and Equation (11), the first term represents potential of heat of a tile with placed CLB, and *Distance_to_CP* is distance between this tile to cool point; therefore, the tiles with higher potential of heat will place to close to cool point; moreover, since we take the pins activity into consideration, the nets with higher activity will route to close to cool point too because its terminals of bounding box are placed close to cool point. The factor ω provides higher flexibility to this cost, and it must be set to less than 1. The factor λ is also set to less than 1 because vertical dissipating path is more important than lateral dissipating path. The Equation (12)~(14) are used to get the coordinates of cool point.

$$Cost_{HD} = \sum_{i \in placed\ CLB} (\omega \times Power_{CLB}(i) + (1-\omega) \times Activity_{Pins}(i)) \times Distance\_to\_CP(i) \quad (10)$$

$$Distance\_to\_CP(i) = |z(i) - CP_z| + \lambda \times (|x(i) - CP_x| + |y(i) - CP_y|) \quad (11)$$

$$CP_x = \left\lceil \frac{nx}{2} \right\rceil \quad (12)$$

$$CP_y = \left\lceil \frac{ny}{2} \right\rceil \quad (13)$$

$$CP_z = nz \quad (14)$$

3.3.3 Interconnect Power

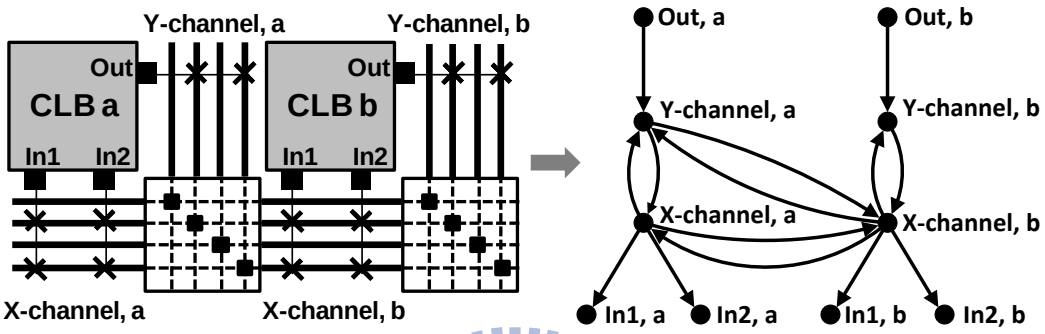
As technology node scales down, the interconnect power becomes dominant the total power, it can contribute 75~85% of total power [25][26]. However, both power uniformity cost and power dissipativity cost may cause the wirelength of some nets longer, and interconnect power will increases. Hence, we want to prevent increasing the interconnect power excessively; we take interconnect power into consideration. As shown in Equation (15), this cost function is the same as Equation (3).

$$Cost_{IP} = \sum_{i \in Net} Activity(i) \times Cost_{Wire}(i) \quad (15)$$

Chapter 4
Routing Algorithm

4.1 Related Works

4.1.1 TPR



(a) FPGA routing architecture (b) Routing-resource graph

Figure 25. Routing-resource graph .

In FPGA routing, a directed graph named *routing-resource graph* is usually used to represent the routing architecture of the FPGA. Each wire segment, TSV and CLB pin becomes a node, and potential connections become edges in this graph as shown in Figure 25.

The routing algorithm in TPR is based on Pathfinder negotiated congestion algorithm [27], and the flow chart is shown in Figure 26. Initially, for each net, the router finds the path with lowest total cost between a net source node and a net sink node in the routing-resource graph. In this step, congestion cost is set to 0; that is, some routing-resources are overused probably. Consequently, the *routing iteration* which ripping-up and re-routing every net in the circuit, is performed until every net employs dedicated routing-resource; before each ripping-up and re-routing, the cost of overusing a routing-resource is increased and updates criticality of each net.

$$Cost(r) = Criticality(n) \cdot Delay(r) + (1 - Criticality(n)) \cdot Congestion(r) \quad (16)$$

The cost of overusing a routing-resource node is shown in Equation (16). The parameter *r* represents a routing-resource, and the parameter *n* represents a net. In this cost function, the timing-critical net will be routed by minimum delay path, while the non-timing-critical net will take a longer, uncongested path, for avoiding the overuse of routing-resource.

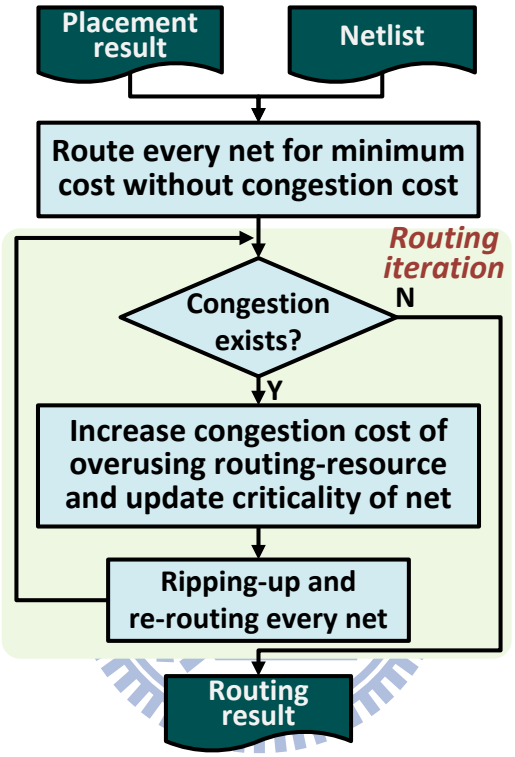
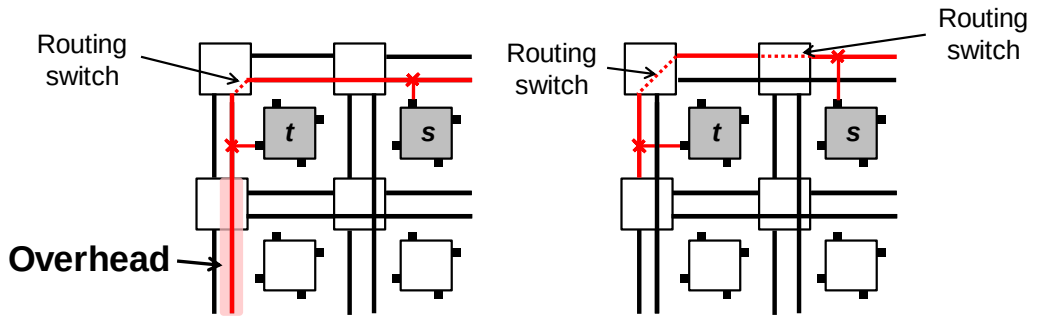


Figure 26. Flow chart of routing algorithm.

4.1.2 3D MEANDER

In TPR routing, in order to minimize delay, the long wire segments are often used to routing because these long wire segments pass less routing switches. Unfortunately, it has a drawback – power overhead, since longer wire segment has larger capacitance, and larger capacitance may consume more interconnect power. Therefore, 3D MEANDER router tries to use shorter wire segments to route the nets with lower criticality because it can reduce the interconnect power and further minimize temperature. As shown in Figure 27, the net in TPR/3D MEANDER routing

result passes 1/2 routing switches and the wirelength is 4/3; thus the 3D MEANDER routing result gets worse delay but the power overhead is minimized.



(a) TPR routing result (b) 3D MEANDER routing result

Figure 27. An example for TPR and 3D MEANDER routing result.

$$Cost(r) = Criticality(n) \cdot Delay(r) + (1 - Criticality(n)) \cdot [Activity(n) \cdot Capacitance(r) + (1 - Activity(n)) \cdot Congestion(r)] \quad (17)$$

The cost of using a routing-resource node is shown in Equation (17), which is modified from Equation (16). In this cost function, the timing-critical net will be routed along minimum delay path, while the non-timing-critical net with high switching activity will be routed by low capacitance wire segments, and the non-timing-critical net with low switching activity is used to avoid the routing-resource congestion.

4.2 Motivation

In 3D MEANDER routing, the non-timing-critical net will be routed by low capacitance wire segments. For example as shown in Figure 28, these six routing results are regarded as the same for 3D MEANDER. However, if we calculate the known power for each tile, and update interconnect power after routing, it can be observed that these six results are totally different for power distribution, and non-uniform power distribution maybe generates hotspots, such as (b), (d) and (f) as shown in Figure 29.

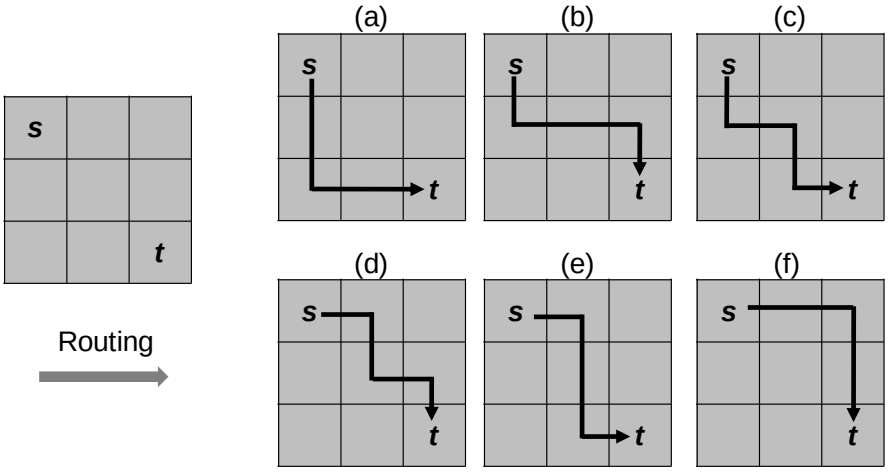


Figure 28. Routing example of 3D MEANDER.

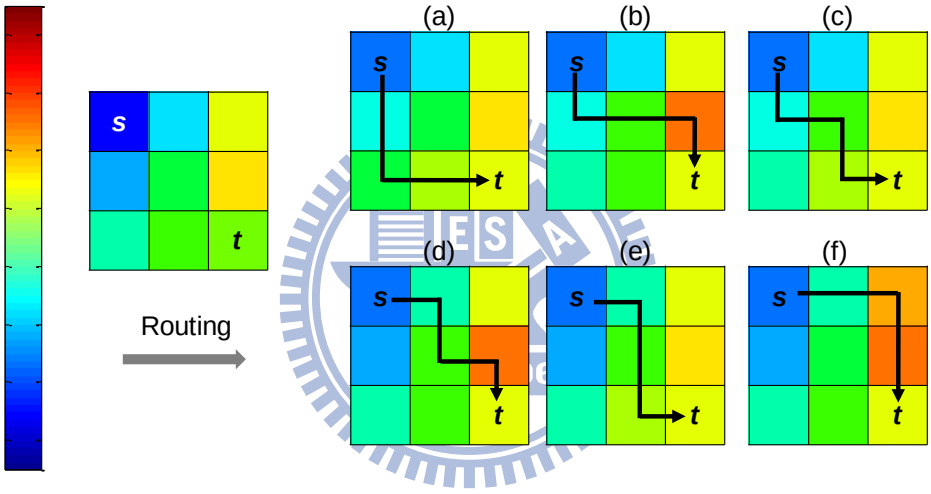


Figure 29. Routing example with power consideration.

Therefore, we want to choose one result with lower probability to generate hotspot regions during our proposed routing algorithm.

4.3 Proposed Algorithm – TherWare

The proposed routing algorithm named TherWare routing, which is based on 3D MEANDER routing. In order to avoid to generate hotspot regions, we perform a simple idea which routes each non-timing-critical net along lower power path; that is, if more than one routing-resource has the same cost, the TherWare routing will find out that which tiles are located by these routing-resources first, and then, it compares

their tile power greedily and choose the smallest tile, and this idea will let the power distribution more uniformly as shown in Figure 30. Notice that the value in each tile represents the tile power, which contains both logic power and known interconnect power, and larger value represents higher power consumption.

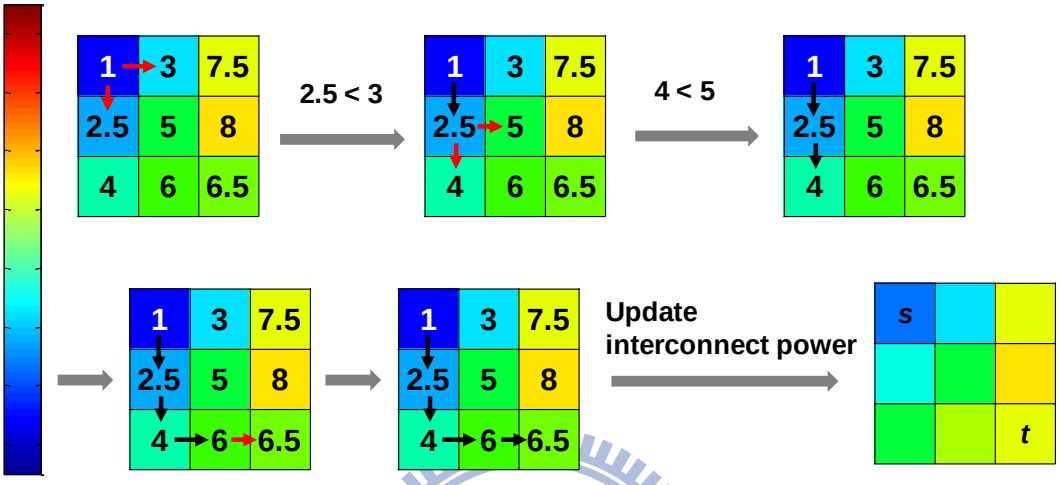


Figure 30. Proposed routing algorithm.

Chapter 5
Experimental Results

5.1 Experimental Environment

The physical settings in our experiments are shown in Table 2. We implement our 3D FPGAs in 45nm technology, the pitch of each TSV is 6μm [1], and area of each tile is 47.8×47.8 μm^2, which is estimated by [17].

Table 2. Physical settings.

Physical Settings	Value
Process technology node	45 nm
TSV Pitch	6 um
Tile Area	47.8×47.8 um^2

The architectural setting in our experiments are shown in Table 3. The settings of CLBs are based on Altera Stratix IV [28]. The horizontal channel width is set to 32 based on Xilinx FPGAs [29]. Each vertical channel also contains 32 TSVs [20]. There are 4 wire segments with different lengths in these 32 wires, L1, L2, L4 and L8. The length of a wire segment is the number of CLBs it spans. There are 12 L1/L2 and 4 L4/L8 wires. In Z direction, each TSV spans one layer only for routability.

Table 3. Architectural settings.

Architectural Settings			Value
CLB (Altera Stratix IV)	**# of inputs of a CLB (I)**		8
	# of LUTs in a CLB (N)		2
	# of inputs of a LUT (K)		6
Channel width	**Both W_X and W_Y (Xilinx)**		32
	W_Z		32
# of wire segments	**X-Y directions**	**(L1, L2, L4, L8)**	(12, 12, 4, 4)
	Z direction	**L1 only**	32

The 20 largest MCNC benchmarks [30] are shown in Table 4, which is sorted by number of CLBs, i.e., problem size. We employ these circuits for our experiments.

Table 4. The MCNC benchmark circuits.

Name	# of IOs	# of Nets	# of CLBs
tseng	174	826	481
ex5p	71	939	529
apex4	27	1073	625
diffeq	103	1116	639
dsip	426	1372	685
misex3	28	1132	689
alu4	22	1216	734
des	501	1505	744
seq	76	1426	847
bigkey	426	1513	908
apex2	41	1537	926
s298	10	1337	1004
elliptic	245	2228	1371
frisc	136	2222	1442
spla	62	2952	1878
pdc	56	3648	2291
ex1010	20	3788	2319
s38584.1	322	5114	3029
s38417	135	5437	3108
clma	130	6379	3869

5.2 Experiment I

In this experiment, we want to observe that how much the TherWare placement and routing improve respectively. Figure 31 shows the experimental flows in experiment I. First, TSV-driven 3D layering [31] is performed. Then, four placement and routing combinations are implemented – i) TPR placement and routing, it is a timing-driven algorithm, and we regard this flow as a baseline, i.e., the other result of flows are normalized to this result of flow. ii) TherWare placement with TPR routing. iii) TPR placement with TherWare routing. iv) TherWare placement and routing. Finally, FG-4 thermal model is used to perform thermal analysis. The power model [32] is used to analyze the switching activity for each net and the power consumption for each CLB. It provides the information for thermal-aware placer and router, after

placement and routing, it calculates every tile power for thermal evaluation.

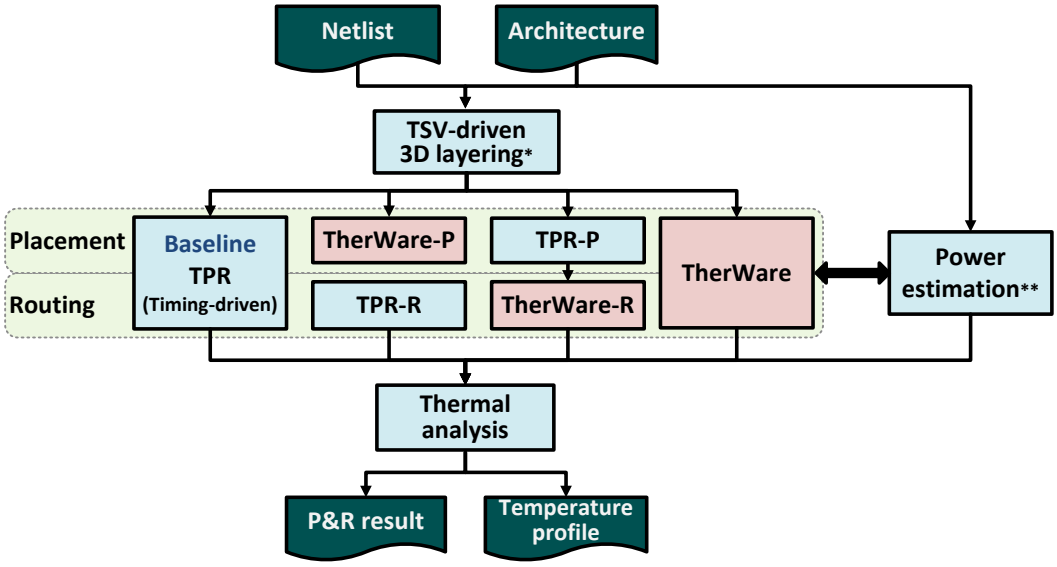


Figure 31. Experimental flows in experiment I.

In addition, the number of layers (nz) is set to 4. The logic utilization is set to 75%, i.e., nx and ny are calculated by Equation (18).

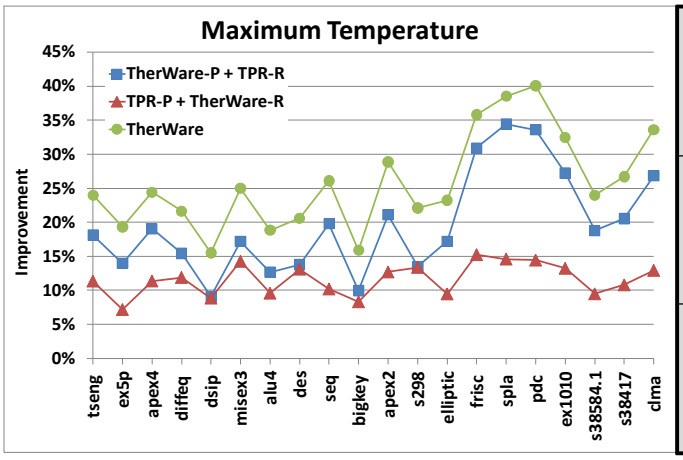
$$nx = ny = \left\lceil \sqrt{\frac{\#\ of\ CLBs}{nz \times utilizatio\ n}} \right\rceil \tag{18}$$

5.2.1 Temperature

Figure 32 shows the improvement of maximum temperature for each benchmark. In average, TherWare has the most improvement which is 25.8%; the improvement of TherWare placement with TPR routing is 19.6%, and TPR placement with TherWare routing is 11.6%. It can be observed that placing the CLBs with higher potential of heat as well at placement stage is more effective for temperature.

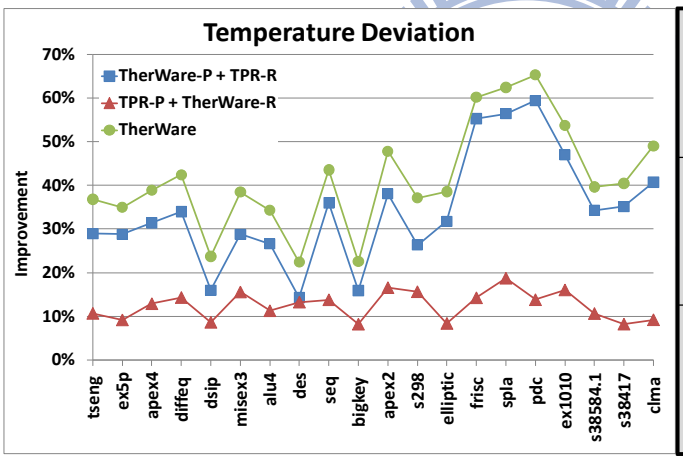
Figure 33 and Figure 34 shows the improvement of temperature deviation and the improvement of maximum temperature gradient, respectively. The maximum temperature gradient is maximum difference of temperature for any two adjacent tiles. In TherWare, both placement and routing are tried to distribute the power uniformly; however, routing must be based on placement result, if the power distribution is very non-uniform after placement, the router will be hard to even the power distribution. It

is the reason that why the improvement of temperature deviation and maximum temperature gradient in TPR placement with TherWare routing is less than that in other flows.



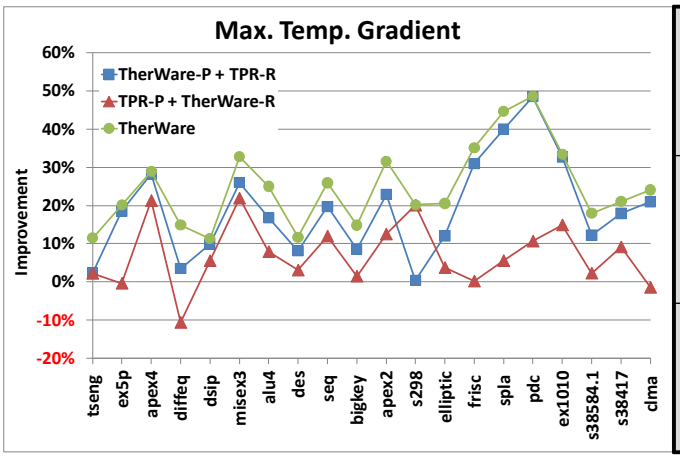
TherWare-P + TPR-R	19.6%
TPR-P + TherWare-R	11.6%
TherWare	25.8%

Figure 32. Improvement of maximum temperature.



TherWare-P + TPR-R	34.3%
TPR-P + TherWare-R	12.5%
TherWare	41.6%

Figure 33. Improvement of temperature deviation.

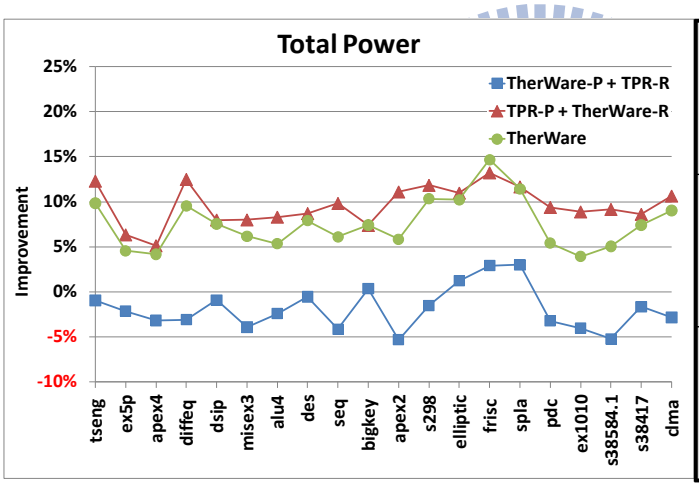


TherWare-P + TPR-R	19.0%
TPR-P + TherWare-R	7.1%
TherWare	24.7%

Figure 34 Improvement of maximum temperature gradient.

5.2.2 Total Power

Figure 35 shows the improvement of total power for each benchmark. Though the proposed placement and routing focus on power distribution, still the improvements of total power are 9.6% and 7.6% in TPR placement with TherWare routing and TherWare, respectively, because we use 3D MEANDER routing as kernel in TherWare routing. TherWare placement with TPR routing gets negative improvement (i.e., overhead); that is, its total power is larger than baseline, but regarding to the temperature, the improvement of maximum temperature is about 20%, it verifies that power optimization is not sufficient for temperature optimization.



TherWare-P + TPR-R	-1.9%
TPR-P + TherWare-R	9.6%
TherWare	7.6%

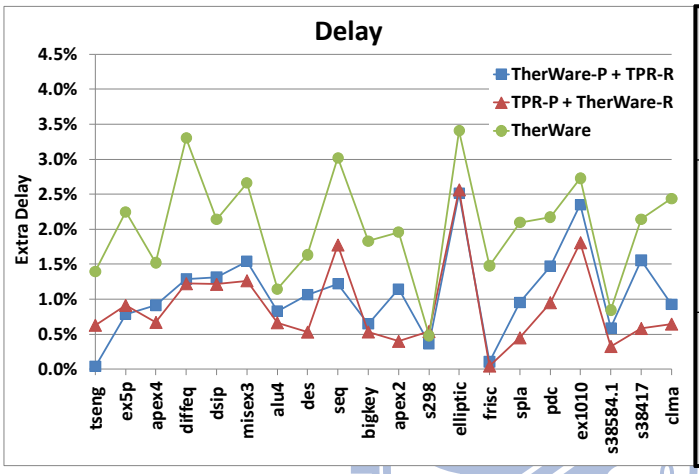
Figure 35. Improvement of total power.

5.2.3 Delay and Runtime

Figure 36 shows the delay overhead for each benchmark. TherWare has 2.0% delay overhead in average even if it takes three ideas into thermal cost function in placement stage, and routes each net along lower power path.

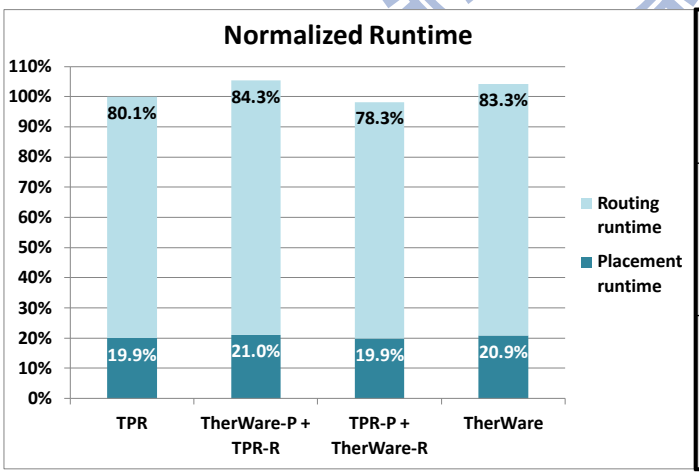
Figure 37 shows the average normalized runtime for each flow, and we separate the total runtime into placement runtime and routing runtime for discussion. The placement runtime is approximately 20% for all flows, and it has a little bit runtime

overhead (~1% of total runtime) in our TherWare placement. Especially, TPR placement with TherWare routing gets negative overhead (i.e., improvement) of routing runtime, this is because we use 3D MEANDER routing as kernel, and the cost function of 3D MEANDER routing is more complex than that of TPR routing, it can avoid the routing-resource congestion easily due to less congestion for non-critical nets, so the number of routing iteration is reduced.



TherWare-P + TPR-R	1.1%
TPR-P + TherWare-R	0.9%
TherWare	2.0%

Figure 36. Delay overhead.



TherWare-P + TPR-R	5.3%
TPR-P + TherWare-R	-1.8%
TherWare	4.2%

Figure 37. Runtime overhead

5.3 Experiment II

In this experiment, we compare the TherWare with related works. The experimental flow is similar to experimental flow in experiment I as shown in Figure 38. Four backend flows with different placement and routing are invoked – i) TPR placement and routing, it is a timing-driven algorithm, and we regard this flow as a baseline, i.e., the other result of flows are normalized to this result of flow. ii) 3D MEANDER placement and routing. iii) Z-tile placement with TherWare routing. iv) TherWare placement and routing. The number of layers is set to 4, and logic utilization is set to 75%, these settings are the same as experiment I.

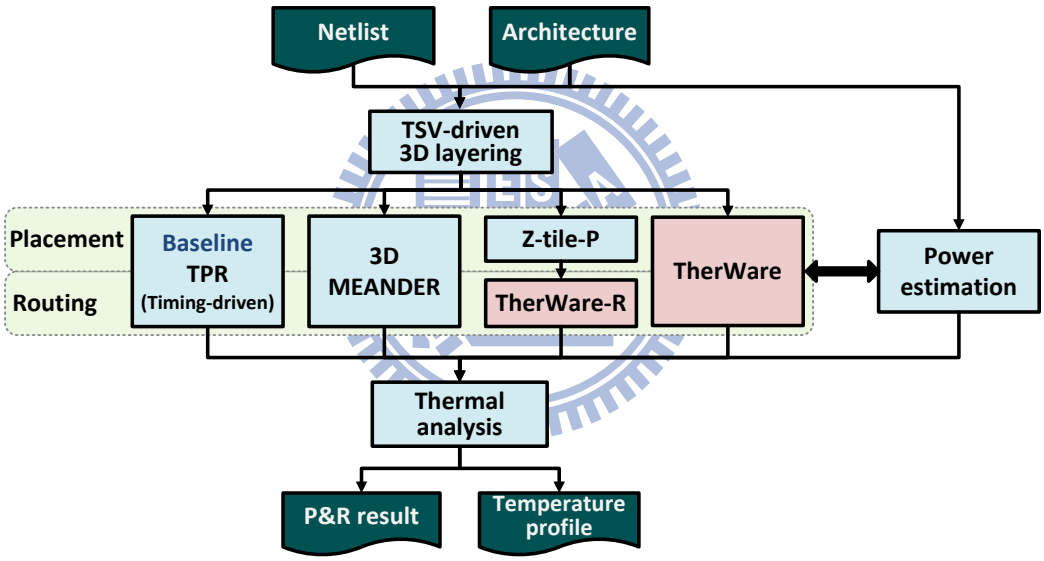


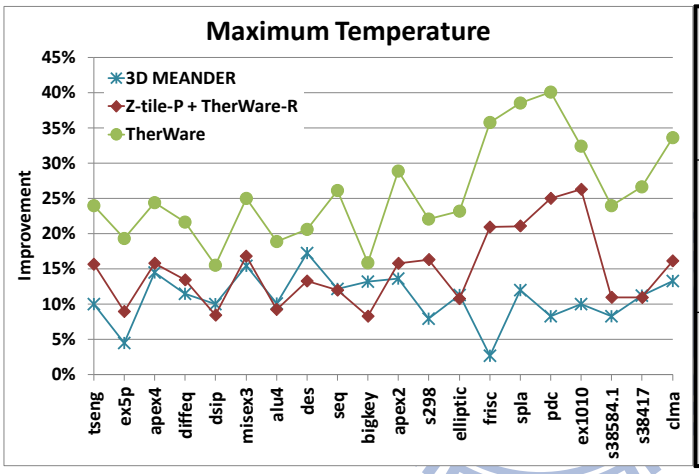
Figure 38. Experimental flows in experiment II.

5.3.1 Temperature

TherWare has the most improvement of maximum temperature among all flows as shown in Figure 39. 3D MEANDER always concentrates on power minimization only, and it does not consider the power distribution; therefore, it has only 10.8% improvement even less than TPR placement with TherWare routing in experiment I. The Z-tile placement with TherWare routing has about 15% improvement because the power distribution is considered in Z-tile placement, while TherWare routing will

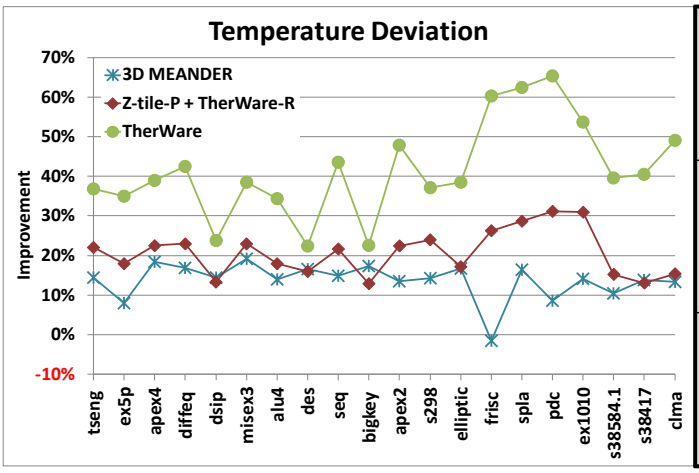
base on its result to route the nets for uniform power distribution too.

Figure 40 and Figure 41 shows the improvement of temperature deviation and maximum temperature gradient, respectively. 3D MEANDER has the smallest improvement due to the same reason – it does not consider the power distribution. Z-tile placement uses the Z-tile thermal model to minimize the maximum temperature; hence, it can improve the temperature deviation and maximum temperature gradient; nevertheless, it cannot outperform TherWare, which distributes the power uniformly all the time.



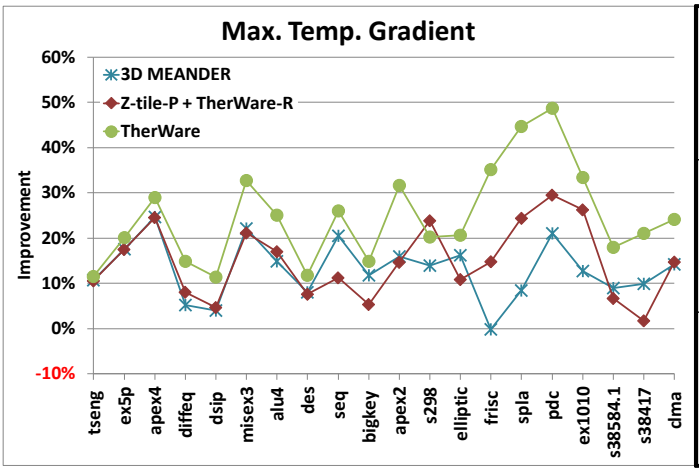
3D MEANDER	10.8%
Z-tile-P + TherWare-R	14.8%
TherWare	25.8%

Figure 39. Improvement of maximum temperature.



3D MEANDER	13.7%
Z-tile-P + TherWare-R	20.7%
TherWare	41.6%

Figure 40. Improvement of temperature deviation.

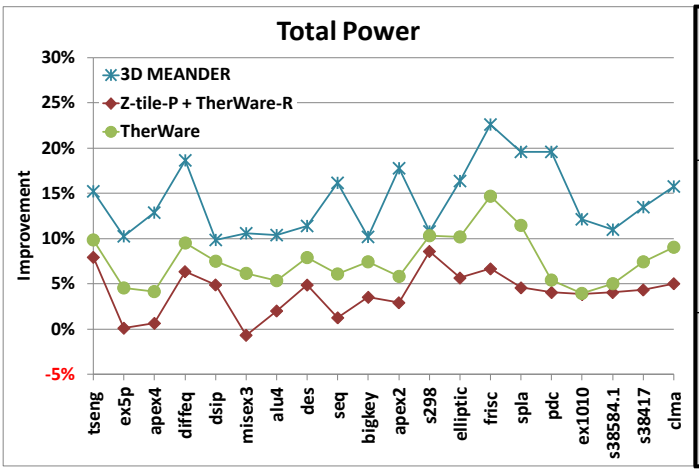


3D MEANDER	13.0%
Z-tile-P + TherWare-R	14.7%
TherWare	24.7%

Figure 41. Improvement of maximum temperature gradient.

5.3.2 Total Power

Figure 42 shows improvement of total power. It is reasonable that the improvement of total power is most significant in the 3D MEANDER, for the concept of 3D MEANDER is simply power minimization. However, this improvement of total power does not guarantee to temperature optimization; it verifies that only power optimization is not enough for temperature optimization again. Furthermore, the Z-tile placement with TherWare routing has less improvement than TherWare because it does not minimize the interconnect power during placement.



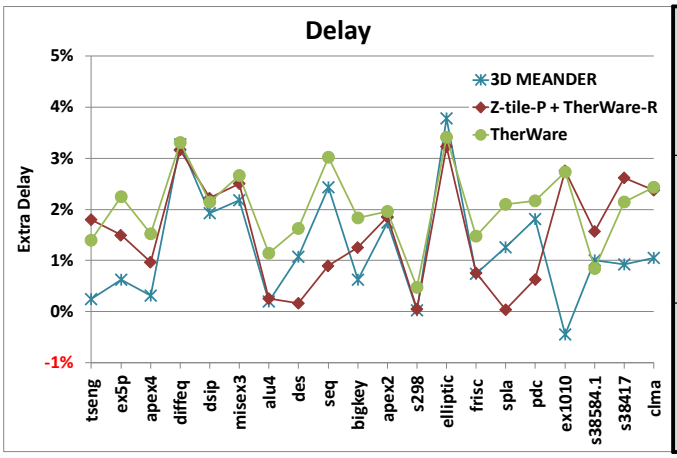
3D MEANDER	14.2%
Z-tile-P + TherWare-R	4.0%
TherWare	7.6%

Figure 42. Improvement of total power.

5.3.3 Delay and Runtime

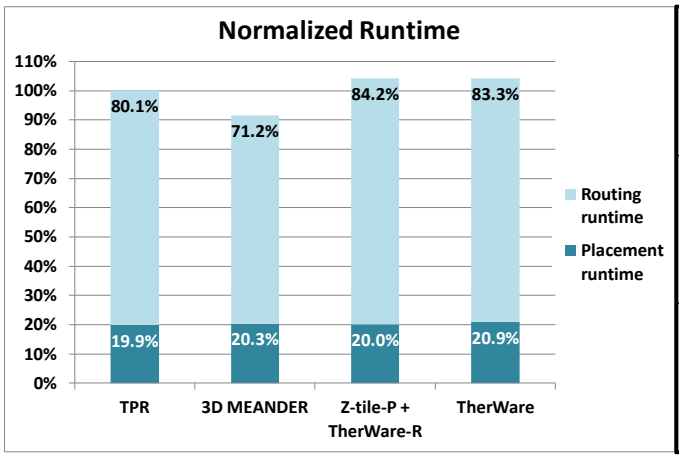
Figure 43 shows the delay overhead for each benchmark. 3D MEANDER and Z-tile placement with TherWare routing has about 1.2% delay overhead in average, while TherWare has 2.0%, which is acceptable to trade-off between temperature and timing optimization.

Figure 44 shows the average normalized runtime for each flow. 3D MEANDER gets negative overhead because its thermal cost is very simple during placement, and the routing has less number of routing iteration. The runtime overhead are 4.2% in Z-tile placement with TherWare routing and TherWare both.



3D MEANDER	1.2%
Z-tile-P + TherWare-R	1.5%
TherWare	2.0%

Figure 43. Delay overhead.



3D MEANDER	-8.5%
Z-tile-P + TherWare-R	4.2%
TherWare	4.2%

Figure 44. Runtime overhead.

5.4 Case Study

In this section, we use apex2 benchmark to observe the effective of four flows for placement and routing in experiment II as various number of layers. The number of layers is 1~8. The logic utilization is set to 75%.

Figure 45 shows the temperature profile at layer 1 of 4-layer design. It can be observed that our TherWare has the smoothest temperature profile significantly.

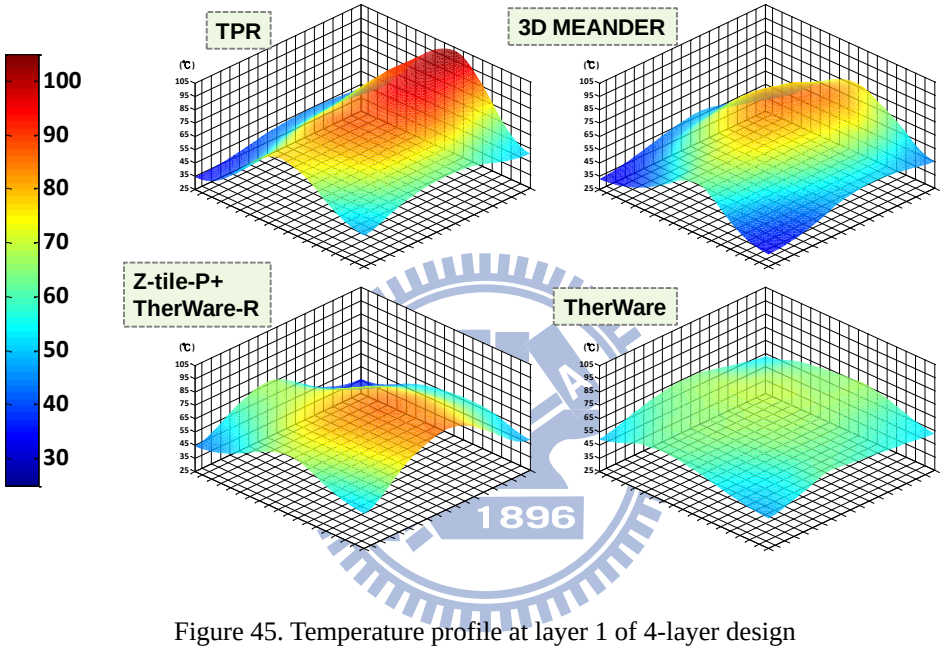


Figure 45. Temperature profile at layer 1 of 4-layer design

The maximum temperature is increased as number of layers increased in all flows as shown in Figure 46. Observably, our TherWare always has the lowest maximum temperature, and it is about 70℃ decreased at a 8-layer design. Similarly, as shown in Figure 47, the temperature deviation grows as number of layers increases in all flows and our TherWare has the lowest deviation also.

Figure 48 shows the delay at 1~8-layer design. The delay decreases as number of layers increases and becomes saturated, mainly because the benefit on shortening global interconnects for 3D designs.

To combine the tradeoff between temperature optimization and impact on timing, our TherWare framework obtains the most benefit to the temperature behavior with

less impact on critical delay, which is especially obvious in a 4-layer design.

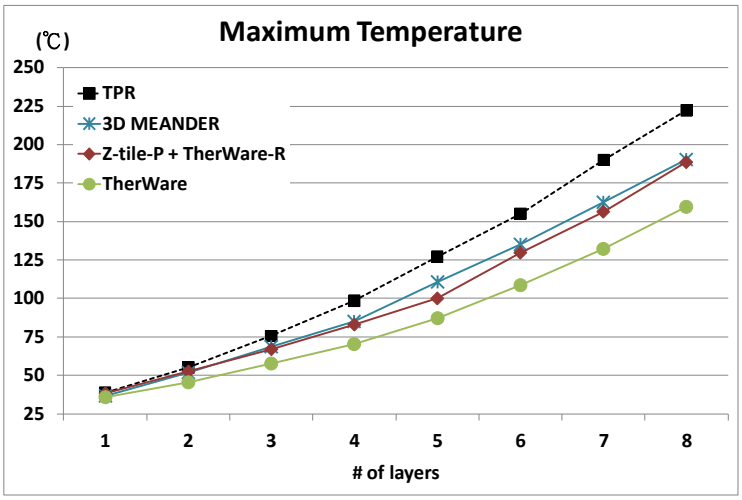


Figure 46. Maximum temperature at 1~8-layer design.

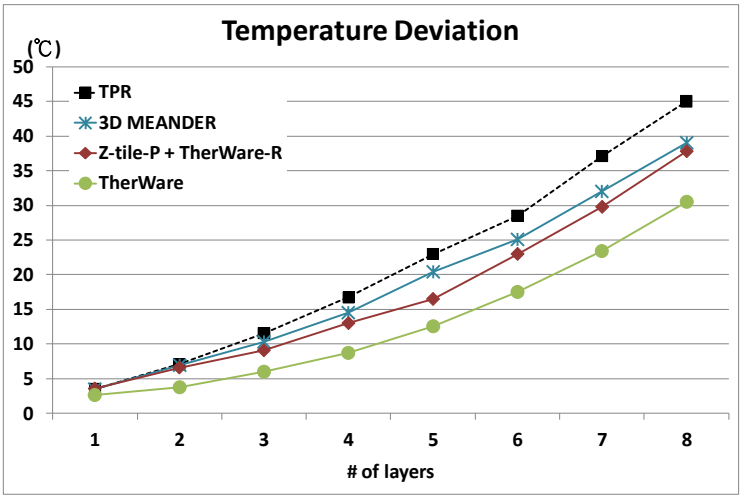


Figure 47. Temperature deviation at 1~8-layer design.

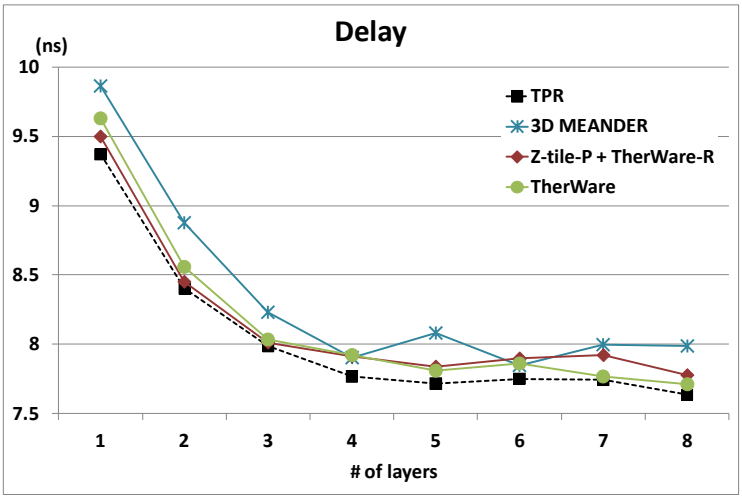


Figure 48. Delay at 1~8-layer design.

Chapter 6
Conclusion

In thesis, we first develop a set of fine-grained thermal resistive models with different granularities for 3D FPGAs, named FG-8, FG-4 and FG-2, respectively. Regarding the finest granularity model – FG-8 as a baseline, the FG-4 is not only accurate but also efficient. For FG-4, the root mean square error is less than 2.5%, and the maximum absolute difference is less than 3.9%. Compared with FG-8, FG-4 also obtains 99.8% correlation and achieves 7.3 times speedup in runtime.

Meanwhile, we also propose a thermal-aware backend framework for 3D FPGAs, named *TherWare*. Three guidelines are integrated in TherWare placement stage: i) power uniformity – keeping power uniformity between several tiles with placed CLB; ii) heat dissipativity – letting the potentially hotter tiles can dissipate heat easily; iii) interconnect power – preventing increasing the interconnect power excessively. Our router allows non-timing-critical nets choosing longer paths with lower power consumption, and such an idea will help distribute power more uniform in 3D FPGA designs.

Table 5. Improvements of TherWare vs. different baseline.

Baseline	Maximum temperature	Temperature deviation	Maximum temperature gradient	Total power	Delay overhead	Runtime overhead
TPR	25.8%	41.6%	24.7%	7.6%	2.0%	4.2%
3D MEANDER	16.6%	31.9%	13.3%	-7.9%	0.8%	14.0%
Z-tile-P + TherWare-R	13.1%	26.9%	11.8%	3.7%	0.5%	-0.3%

The experimental results are summarized in Table 5. TherWare outperforms all related works on temperature, and it has only few overheads in delay and runtime. We conclude that TherWare is the most effective thermal-aware placement and routing for 3D FPGAs up to now.

Reference

[1] International Technology Roadmap for Semiconductor. Semiconductor Industry Association 2005–2010.

[2] A. W. Topol, D. C. La Tulipe, L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Ieong, "Three-dimensional integrated circuits," IBM J. of Research and Development, vol. 50, no. 4.5, pp. 491–506, Jul. 2006.

[3] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: a novel chip design for improving deep submicron interconnect performance and systems-on-chip integration," Proc. IEEE, vol. 89, no. 5, pp. 602–633, May 2001.

[4] R. Tummala and V. Madisetti, "System on chip or system on package?" IEEE Design & Test of Computers, vol. 16, no. 2, pp. 48–56, Apr.–Jun. 1999.

[5] P. H. Shiu and K. S. Lim, "Multi-layer floorplanning for reliable system-on-package," Proc. Int'l Symp. Circuits and System, pp. 23–26, 2004.

[6] S. Spiesshoefer, Z. Rahman, G. Vangara, S. Polamreddy, S. Burkett, and L. Schaper, "Process integration for through-silicon vias," J. of Vacuum Science and Technology A, vol. 23, no. 4, pp. 824–829, Jul. 2005.

[7] SOCcentral. [Online]. Available: http://www.soccentral.com

[8] S. Das, A. P. Chandrakasan, and R. Reif, "Calibration of rent's rule models for three-dimensional integrated circuits," IEEE Trans. Very Large Scale Integration Systems, vol. 12, no. 4, pp. 359–366, Apr. 2004.

[9] A. Rahman and R. Reif, "System-level performance evaluation of three-dimensional integrated circuits," IEEE Trans. Very Large Scale Integration Systems, vol.8, no.6, pp. 671–678, Dec. 2000.

[10] S. Das, A. Fan, K. Chen, C. S. Tan, N. Checka, and R. Reif, "Technology, performance, and computer-aided design of three-dimensional integrated circuits," Proc. Int'l Symp. Physical Design, pp. 108–115, 2004.

[11] I. Kaya, S. Salewski, M. Olbrich, and E. Barke, "Wirelength reduction using 3D physical design," Int'l Workshop Integrated Circuit System Design, pp. 453–462, 2004.

[12] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, and P. D. Franzon, "Demystifying 3D ICs: the pros and cons of going vertical," IEEE Design & Test of Computers, vol. 22, no. 6, pp. 498–510, Nov.–Dec. 2005.

[13] I. Loi, S. Mitra, T. H. Lee, S. Fujita, and L. Benini, "A low-overhead fault

tolerance scheme for TSV-based 3D network on chip links," Proc. Int'l Conf. Computer-Aided Design, pp. 598–602, 2008.

[14] C. Ababei, H. Mogal, and K. Bazargan, "Three-dimensional place and route for FPGAs," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 6, pp. 1132–1140, Jun. 2006.

[15] S. Im and K. Banerjee, "Full chip thermal analysis of planar (2D) and vertically integrated (3D) high performance ICs," Technical Dig. Int'l Electron Devices Meeting, pp. 727–730, 2000.

[16] T. Y. Chiang, S. J. Souri, C. O. Chui, and K. C. Saraswat, "Thermal analysis of heterogeneous 3D ICs with various integration scenarios," Technical Dig. Int'l Electron Devices Meeting, pp. 681–684, 2001.

[17] V. Betz, J. Rose, and A. Marquardt, Architecture and CAD for deep-submicron FPGAs, Kluwer Academic Publishers, 1999.

[18] P. Wilkerson, A. Raman, and M. Turowski, "Fast, automated thermal simulation of three-dimensional integrated circuits," Int'l Society Conf. on Thermal Phenomena, vol. 1, pp. 706–713, Jun. 2004.

[19] W. Huang, "HotSpot - A chip and package compact thermal modeling methodology for VLSI design," PhD Thesis, ECE, University of Virginia, 2007.

[20] C.-I Chen, B.-C. Lee, and J.-D. Huang, "Architectural exploration of 3D FPGAs towards a better balance between area and delay," Proc. Design, Automation & Test in Europe Conf. and Exhibit., pp. 587–590, 2011.

[21] J. Jaffari and M. Anis, "Thermal-aware placement for FPGAs using electrostatic charge model," Proc. Int'l Symp. on Quality Electronic Design, pp. 666–671, 2007.

[22] S. Im, N. Srivastava, K. Banerjee, and K. E. Goodson, "Thermal scaling analysis of multilevel Cu/Low-k interconnect structures in deep nanometer scale technologies," Proc. Int'l VLSI Multilevel Interconnect Conf., pp. 525–530, 2005.

[23] K. Siozios, V. F. Pavlidis, and D. Soudris, "A software-supported methodology for exploring interconnection architectures targeting 3D FPGAs," Proc. Design, Automation & Test in Europe Conf. and Exhibit., pp. 172–177, 2009.

[24] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," Proc. Int'l Conf. Computer-Aided Design, pp. 306–313, 2004.

[25] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power efficient FPGAs," Proc. Int'l Symp. on Field Programmable Gate Arrays, pp. 175–184, 2003.

[26] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics

of field programmable gate arrays," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 11, pp. 1712–1724, Nov. 2005.

[27] L. McMurchie, and C. Ebeling, "Pathfinder: A Negotiation-Based Performance-Driven Router for FPGAs," Proc. Int. Sym. on Field-Programmable Gate Arrays, pp.111–117, 1995.

[28] Altera. [Online]. Available: http://www.altera.com/

[29] Xilinx. [Online]. Available: http://www.xilinx.com/

[30] S. Yang, "Logic synthesis and optimization benchmarks user guide," Technical Report 1991-IWLS-UG-Saeyang, Microelectronics Center of North Carolina, 1991.

[31] Y-S. Huang, Y.-H. Liu, and J.-D. Huang, "Layer-Aware Design Partitioning for Vertical Interconnect Minimization," Proc. IEEE Computer Society Annual Symp. on VLSI, pp. 144–149, 2011.

[32] Kara K. W. Poon, Steven J. E. Wilton , and Andy Yan, "A detailed power model for field-programmable gate arrays," ACM Trans. on Design Automation of Electronic Systems, vol. 10, no. 2, pp. 279–302, Apr. 2005.