UC Irvine
UC Irvine Electronic Theses and Dissertations

Title
Temperature-Aware Design for SoCs using Thermal Gradient Analysis

Permalink
https://escholarship.org/uc/item/8979k9fc

Author
Shin, Jun Yong

Publication Date
2015

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Temperature-Aware Design for SoCs using Thermal Gradient Analysis

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Jun Yong Shin

Dissertation Committee:
Professor Nikil Dutt, Chair
Professor Fadi Kurdahi
Associate Professor Ahmed Eltawil

2015

Chapter 2 © 2015 IEEE ESL
Chapter 3 © 2015 IEEE ISQED
Chapter 4 © 2013 IEEE ISQED
All other materials © 2015 Jun Yong Shin

DEDICATION

To

my parents

who have

supported me unconditionally

and

believed in me all the time with love

TABLE OF CONTENTS

Page

LIST OF FIGURES vi

LIST OF TABLES ix

ACKNOWLEDGMENTS x

CURRICULUM VITAE xi

ABSTRACT OF THE DISSERTATION xii

CHAPTER 1. Introduction 1

1.1. Thermal issues in Nano CMOS era 1

1.1.1. Reliability 2

1.1.2. Performance 5

1.1.3. Power consumption 8

1.2. Dynamic Thermal Management (DTM) 13

1.2.1. Hardware-based DTM 14

1.2.2. Software-based DTM 15

1.3. Thermal sensors 17

1.3.1. Sensor allocation: hotspot monitoring 18

a) Uniform allocation 18

b) Recursive bisection 19

c) Temperature-aware *k*-means clustering 20

1.3.2. Sensor allocation: full-chip profile reconstruction 21

a) Energy-aware distribution 21

b) Statistical approach 23

c) Sensor-assisted power estimation 23

1.4. Dissertation overview 25

1.5. Application to a chip design 27

CHAPTER 2. On-Chip Temperature Estimation using Multiple Virtual Sensors 31
2.1. Motivation 31
2.2. Related work 32
2.3. Cooperative temperature estimation 33
2.3.1. Modeling 33
2.3.2. Calibration 35
2.3.3. Cooperative estimation 37
a) A sensor group 37
b) Multiple virtual thermal sensors 38
c) Switching virtual sensors 38
d) Prediction 39
2.4. Experimental results 40
2.5. Summary 44
CHAPTER 3. Thermal Sensor Allocation for SoCs Based on Temperature Gradients 46
3.1. Motivation 46
3.2. Related work 47
3.3. Thermal sensor allocation 48
3.3.1. Reference thermal profile generation 48
3.3.2. Edge detection and object labeling 50
3.3.3. Initial sensor allocation 51
a) Sensor points 51
b) Sensor candidates 53
3.3.4. Additional sensor allocation 55
3.3.5. Geometrical framework 56
3.4. Thermal profile reconstruction 57
3.5. Experimental results 59
3.6. Summary 67
CHAPTER 4. Vision-inspired Global Routing for Enhanced Performance and Reliability 68
4.1. Motivation 68
4.2. Related work 69
4.3. Vision-inspired global routing 70

4.3.1. Overall flow 70
4.3.2. Peak and valley detection 72
4.3.3. Solution space expansion 75
4.3.4. Reliability and performance metric 76
4.4. Experimental results 78
4.5. Summary 81
CHAPTER 5. Conclusion 83
5.1. Practical design issues 85
5.2. Improvement of the proposals 87
5.3. Directions for future research 90
REFERENCES 93

LIST OF FIGURES

Page

Figure 1.1. Examples of non-uniform thermal profiles [87] [86] 2

Figure 1.2. Trend in MTF as a function of temperature 4

Figure 1.3. Trend in total interconnect length on a chip [14] 5

Figure 1.4. Trend in interconnect delay [15] 6

Figure 1.5. Trend in power density as a function of gate length [85]... 10

Figure 1.6. Trend in total chip power: portable consumer Systems on Chip (SoC) [88] 11

Figure 1.7. Trend in total chip power: stationary consumer Systems on Chip (SoC) [88]........... 12

Figure 1.8. (a) Thread selection when the integer register file is thermally critical [34], (b) Temperature-aware task scheduling for MPSoC [36] 16

Figure 1.9. Recursive bisection based thermal sensor allocation [42].. 19

Figure 1.10. Energy-aware thermal sensor allocation [48]... 22

Figure 1.11. Thermal profile estimation based on sensor-assisted power estimation [56]........... 24

Figure 1.12. Application of propose methods to a chip design 27

Figure 2.1. A RO-based on-chip thermal sensor.. 33

Figure 2.2. Variations in frequency outputs of RO-based thermal sensors due to process variation and environmental uncertainties 35

Figure 2.3. Temperature reading errors of 1,000 sensors due to variations: (a) with single-point calibration at 63℃, and (b) with two-point calibration at 45℃ and 82℃......................... 36

Figure 2.4. Ideal error bounds in case of four sensors at one location: (a) single-point calibration at 36, 54, 73, 91℃, respectively, (b) two-point calibration at (32, 41), (50, 59), (68, 77), (86, 95)℃, respectively 37

Figure 2.5. Temperature readings of four sensors: (a) with single-point calibration, (b) with two-point calibration 41
Figure 2.6. Estimation results (top) and corresponding errors (bottom) in case of four sensors: single-point calibration (left), and two-point calibration (right) 42
Figure 2.7. Maximum absolute errors as a function of the number of sensors 44
Figure 2.8. RMSE and its reduction rates as a function of the number of sensors 44
Figure 3.1. Steps for thermal sensor allocation and full-chip profile reconstruction 49
Figure 3.2. An example of reference thermal profiles of a chip 49
Figure 3.3. (a) Edge detection, (b) object labeling and analysis 50
Figure 3.4. Sensor point allocation 52
Figure 3.5. Sensor candidate allocation 54
Figure 3.6. (a) Sensor points and sensor candidates (including pseudo candidates), (b) k-means clustering ($k = 6$) on sensor candidates 55
Figure 3.7. (a) 9 thermal sensors (3 from sensor points and 6 from sensor candidates) (b) geometrical framework composed of 32 nodes 57
Figure 3.8. Transform-based reconstruction 58
Figure 3.9. DCT matrix generation 59
Figure 3.10. Sensor allocation and profile reconstruction results when the number of sensors is set to four (or $k = 1$) (a) sensor allocation result, (b) profile to be reconstructed, (c) reconstruction: DCT, (d) reconstruction: regression 60
Figure 3.11. RMSE as a function of the number of sensors 61
Figure 3.12. Absolute error at the hottest spot 62
Figure 3.13. Reference thermal profile of a new chip 63

Figure 3.14. (a) Edge detection, (b) object labeling and analysis of a new chip 64
Figure 3.15. (a) Sensor points and sensor candidates of a new chip, (b) *k*-means clustering on sensor candidates (k = 3) of a new chip 65
Figure 3.16. (a) 9 thermal sensor nodes of a new chip, (b) geometrical framework composed of 105 nodes to which temperature values are assigned 65
Figure 3.17. RMSE comparison among three methods: considering all 4096 nodes, and considering 1024 hot nodes only 66
Figure 4.1. Overall flow of vision-inspired global router 71
Figure 4.2. HotSpot [76]-generated thermal profile 72
Figure 4.3. Gradient at each node on a grid of 64 by 64 nodes 73
Figure 4.4. Peak and valley areas: identifying nodes with small gradient magnitude 74
Figure 4.5. Peak and valley areas: classification 74
Figure 4.6. Solution space expansion by adding a limited number of internal nodes 75
Figure 4.7. Routing of a net given in Figure 4.6 77
Figure 4.8. Reduction rates: comparison with a conventional router 80
Figure 4.9. Trend in reduction rates: our router in comparison with TAGORE [58] 80
Figure 4.10. Delay reduction: comparison with a conventional router 81
Figure 5.1. Practical design issues: sensor accuracy and sensor allocation 86
Figure 5.2. Calibration points of virtual sensors: (a) uniform error bounds over the wide temperature range, (b) non-uniform error bounds over the wide temperature range, (c) uniform error bounds over the narrow temperature range 88
Figure 5.3. Improvement of the proposals: sensor allocation 89

LIST OF TABLES

Page

Table I. Random variables for Monte Carlo simulations 40

Table II. Maximum absolute error and RMSE of estimation for 20 profiles 43

Table III. Comparison in the number of nodes in hotspots and delay reduction 79

ACKNOWLEDGMENTS

It was a long journey. Sometimes, I was even not sure whether or not I was following a right path to my destination. Without proper guidance and continued support from people who cared about me, I would have lost my way a long time ago.

Firstly, I would like to express my deepest appreciation to Professor Nikil Dutt, the committee chair, who has consistently shown me what it is to be a great scholar and how the researches should be performed in an era when most new ideas and new technologies easily become obsolete in a blink of an eye. I also would like to express my hearty appreciation to Professor Fadi Kurdahi who was always willing to help me whenever I was in need of anything and led me in the right direction in performing research. I also would like to thank Professor Ahmed Eltawil for letting me have a detailed and extensive review of my work.

Secondly, I would like to thank Professor Hemantha K. Wickramasinghe, Professor Glenn Healey, and Professor Mark Bachman who all showed me what a good professor should be like in class, and how to become a good one to understand the needs of students and to help students expand their knowledge on a subject without any limitation.

Thirdly, I would like to thank all staff in the department of Electrical Engineering and Computer Science who helped me whenever I had difficulties in figuring out what to do in numerous perplexing situations. In addition, I should not forget about great help from Dr. Magdy Abadir and Dr. Aseem Gupta during my internship at Freescale semiconductor.

Finally, I'd like to thank IEEE, the Institute of Electrical and Electronics Engineers, for permission to include copyrighted materials as part of my dissertation. Chapter 2 is mainly based on the material that appears in IEEE Embedded Systems Letters (ESL) in 2015, and chapter 3 is based on the material in IEEE ISQED 2015. Chapter 4 is from the material that appears in IEEE ISQED 2013. The co-authors listed in those publications directed and supervised research that forms the basis of this dissertation.

CURRICULUM VITAE

Jun Yong Shin

1995	B. S. in Electronics Engineering, *Cum Laude* Hanyang University, Seoul, South Korea
1997	M. S. in Electronics Engineering, Hanyang University, Seoul, South Korea
1997 -	Assistant Manager, Hyundai Motor Company R&D center, Jeonju, South Korea
2000 -	Senior Research Engineer, LG Electronics Inc. CDMA handsets R&D center, Seoul, South Korea
2015	Ph.D. in Electrical and Computer Engineering, University of California, Irvine

FIELD OF STUDY

Embedded system design: temperature-aware design

RELATED PUBLICATIONS

Jun Yong Shin, Fadi Kurdahi, and Nikil Dutt, "Thermal Sensor Allocation for SoCs Based on Temperature Gradients," in *ISQED*, 2015

Jun Yong Shin, Fadi Kurdahi, and Nikil Dutt, "Cooperative On-Chip Temperature Estimation Using Multiple Virtual Sensors," *IEEE Embedded Systems Letters*, 2015

Jun Yong Shin, Nikil Dutt, and Fadi Kurdahi, "Vision-inspired Global Routing for Enhanced Performance and Reliability," in *ISQED*, 2013

ABSTRACT OF THE DISSERTATION

Temperature-Aware Design for SoCs using Thermal Gradient Analysis

By

Jun Yong Shin

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2015

Professor Nikil Dutt, Chair

Over the last few decades, chip performance has increased steadily due to continuous and aggressive technology scaling. However, it leaves chips quite vulnerable to several issues at the same time. High power densities in some particular areas spread across a chip might result in hotspots and thermal gradients, and these can lead to permanent damage to the chip and also can reduce the reliability of the entire system using the chip. As a result, a large number of dynamic thermal management (DTM) solutions have been proposed in recent years for use in multi-core architectures, and accurate temperature information over the entire chip area has become indispensable especially for fine-grain DTM solutions. Naturally, on-chip thermal sensors came to play an important role in providing accurate information on the thermal distribution of a chip, but there still remain some issues regarding the allocation of on-chip thermal sensors. Due to power, die area, and routing issues, it is preferable to limit the total number of on-chip thermal sensors on a die. Their placement also needs to be considered carefully in order to increase the accuracy of full-chip thermal profile reconstruction, especially when just a small number of thermal sensors can be deployed. In addition, it would be preferable to have some way to

improve the reading accuracy of low power, small-sized on-chip thermal sensors that usually tend to have very limited accuracy in temperature readings.

In this work, an issue will be firstly addressed regarding how to improve the reading accuracy of a low power, small-sized on-chip thermal sensor such as Ring-Oscillator (RO) based sensors at runtime on a software level. Secondly, a question of how to allocate a proper number of thermal sensors on a die in order to get the accurate full-chip scale temperature information on the run is addressed. Additionally, a temperature-aware routing method for global interconnects to minimize the signal propagation delay and also to reduce the probability of chip failure due to electromigration is presented at the end.

CHAPTER 1. Introduction

1.1. Thermal issues in Nano CMOS era

In the past few decades, downscaling of chips has been playing a key role in reducing the power consumption and also in increasing the performance of chips. The size of gate length, gate oxide thickness, and other design parameters will keep shrinking for a while in the future even though the shrinking trend will be slowed down slightly due to some technology difficulties such as an escalated cost and process complexity in lithography, the reliability of extremely fine interconnects, etc. Downscaling of chips or the continued shrinkage in gate length has naturally increased the power density of chips. Resulting high temperature of chips became one of the biggest issues in chip design, and those thermal issues are becoming more problematic with aggressive technology scaling. In extreme cases, some parts of a chip can be burned out leading to chip failure in the end; thermal runaway, which is caused by positive feedback between increased leakage current and high temperature, can be thought of as one example of such a case. In addition, as we put a lot of heterogeneous components on a chip, the thermal distribution of a chip tend to become non-uniform, i.e., some parts of a chip are hotter than the others due to different processing tasks in different parts of a chip. Implementing multiple cores instead of increasing the clock frequency of a single core became a trend in processor design as a way of alleviating the burden of high power consumption and enormous heat generation [1], and this trend also plays a role in making the thermal distribution non-uniform over a chip to some extent. Especially when thread mapping

among the multiple cores is not well-balanced, non-uniform thermal distribution can become a lot worse, resulting in multiple localized temperature maxima, which are usually termed hotspots [2]. According to [3], [4], and [5], temperature within a chip can vary as much as 50℃ across a die, and examples of this non-uniform thermal distribution are given in Figure 1.1.

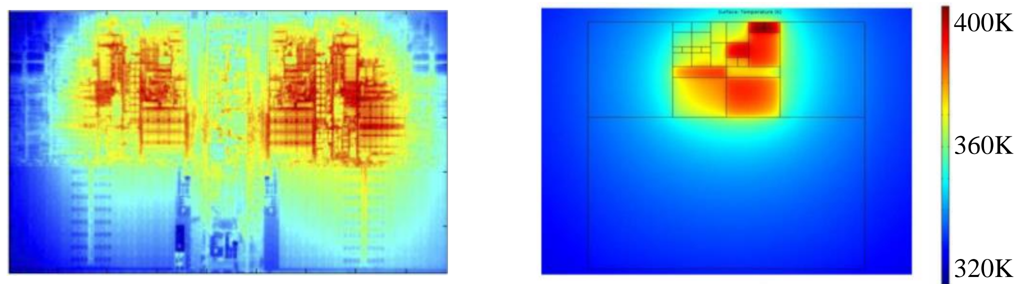


Figure 1.1. Examples of non-uniform thermal profiles [87] [86]

Hotspots and thermal gradient may result in various kinds of issues: reduced reliability of a chip due to electromigration [6], timing failure or communication error between functional blocks in a chip due to increased clock skews, higher cost than before for cooling solutions such as heavy cooling fans, heat sinks, etc. [7]

1.1.1. Reliability

One of the serious issues that can be caused by high operating temperatures and non-uniform thermal distribution over a die is the reduction in the reliability of interconnects and the resulting short life expectancy of a chip due to electromigration [6]. Electromigration is the result of momentum transfer from the collision between electrons and the atoms forming the lattice of the material, and it can cause void or hillock formation along the metal lines in

extreme cases [6]. With CMOS (Complementary Metal-Oxide Semiconductor) technology scaling, the reliability and the life expectancy of interconnects in a chip are becoming more susceptible to electromigration than before. Black's equation [6] or its modified equation [8] given below have been widely used as a way of modeling and predicting the Mean Time to Failure (MTF) of interconnects subjected to electromigration:

$$\mathrm{MTF} = \frac{A}{J^n} e^{\frac{E}{kT}} \qquad (1.1)$$

In this equation, A is a constant that is determined by the material properties and the geometry of the interconnects, J is the current density, n is a scaling factor that is to be determined experimentally, E is the thermal activation energy depending on the used material, k is the Boltzmann's constant, and T is the absolute temperature of the metal in the unit of Kelvin. The current density exponent n is usually set to a value between one and two, and it depends on the failure mechanism [9]; a value close to one characterizes well the failure due to void growth [10]; a value close to two represents the failure due to void nucleation quite well [11]. In the equation, two dominant factors determining the MTF of interconnects are the current density J and the temperature T. As CMOS technology scales down, the current density of interconnects generally increases [12], so the life expectancy of interconnects will decrease. To make it worse, the MTF decreases exponentially with respect to the temperature of interconnects. For example, when the temperature of an interconnect changes from 45°C to 65°C, the life expectancy of the interconnect is reduced by 70% roughly, and the chip will fail

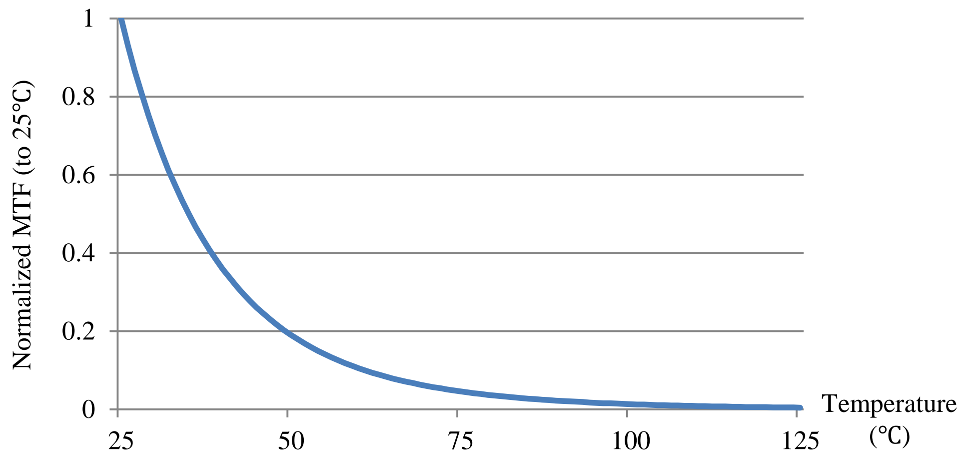


Figure 1.2. Trend in MTF as a function of temperature

much sooner than before if we design chips in a traditional way without proper consideration on thermal issues and adequate cooling solutions. The trend in MTF, which is normalized so that the MTF at 25°C is to be one, is given in Figure 1.2 as a function of temperature.

As process scaling develops further, the top metal layers get closer to the substrates, and this will further intensify the impact of thermal gradients of substrates on the thermal profile of interconnects [13]; thus, the reliability or the MTF of interconnects decreases exponentially with the increase in the temperature of substrates. In order to improve the reliability or the MTF of interconnects, it becomes indispensable to manage the thermal distribution of a chip dynamically and also to consider the thermal distribution of substrates during chip design or interconnect design stage so that we can avoid hot regions or hotspots on the substrates for the routing.

1.1.2. Performance

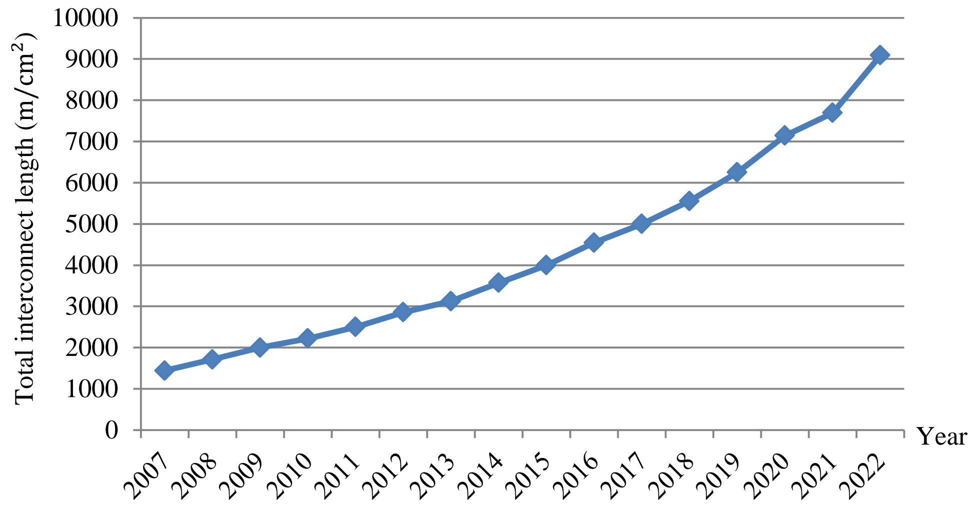


Figure 1.3. Trend in total interconnect length on a chip [14]

As CMOS technology scales down, the gate delay, i.e., the delay of active devices in a chip, has been reduced quite successfully. On the contrary, the global interconnect delay continuously increased, and as a result, the interconnect delay already became more dominant than the gate delay. The total length of interconnects in a chip is expected to reach 9,091 m/cm^2 in the year of 2022 as can be seen in Figure 1.3 [14], and the gap between gate delay and interconnect delay will keep increasing with CMOS technology scaling as given in Figure 1.4 [15]. Rapid increase in interconnect delay makes the timing closure a lot harder than before, and it usually renders or forces the reduction in clock frequency, i.e., the performance limitation in order to prevent any timing-related issues. To make it worse, interconnect delay depends not only on the length of interconnects but also on the thermal distribution of substrates, over which interconnects pass, especially in deep submicron eras. Traditional routing algorithms just use the length of interconnects as a metric for the signal delay,

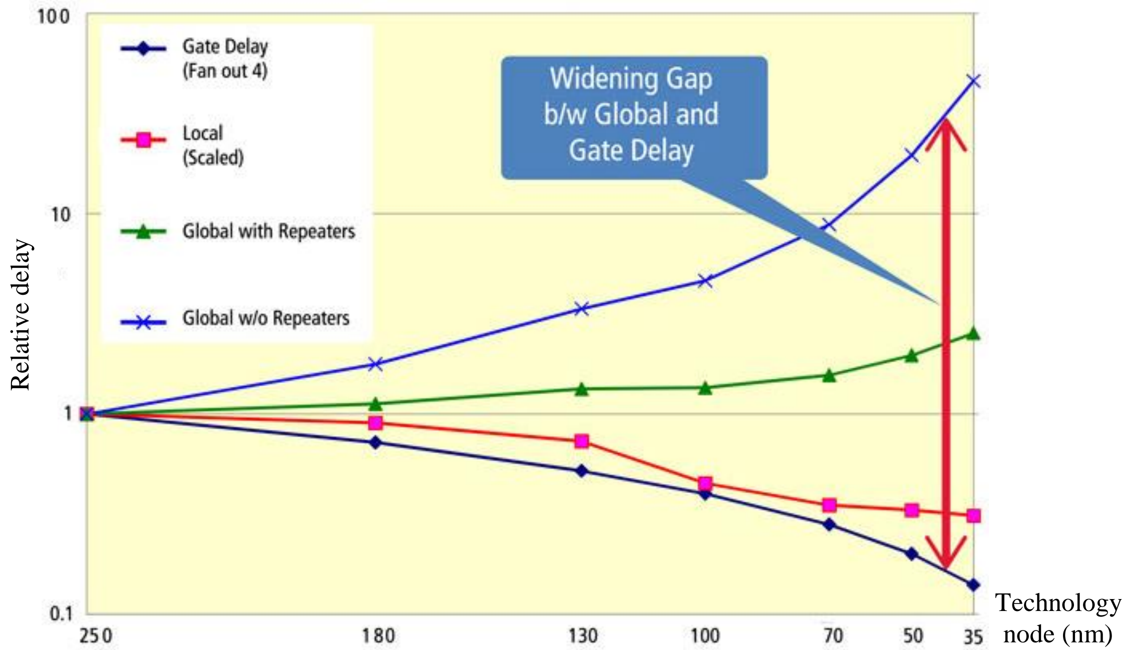


Figure 1.4. Trend in interconnect delay [15]

assuming that the resistivity within interconnects stays constant and uniform, and the non-uniform thermal distribution on the substrates doesn't affect the interconnect delay. These assumptions are not valid any longer in deep submicron technologies; thus, we need to consider both the length and the non-uniform thermal distribution of interconnects, especially for global routing. That is to say, we need to consider thermal distribution during the interconnect design stage in order to prevent any unexpected temperature-related timing failure or to prevent further limitation on performance due to temperature.

According to [16], the temperature within an interconnect for a given substrate temperature can be expressed as follows:

$$T(x) = T_{sub}(x) + \frac{\theta}{\lambda^2}\left(1 - \frac{\sinh \lambda x + \sinh \lambda (L - x)}{\sinh \lambda L}\right) \qquad (1.2)$$

In above equation, T_{sub} is the temperature of the substrates beneath the interconnect, L is the length of the interconnect, and θ and λ are constants for a chosen metal layer in a specific technology node. Both constants depend on the thermal conductivity of the metal and insulator, their geometries, and also on the current density and the resistivity of the interconnect. It is quite well known that the electrical resistance of metal has a linear relationship with its temperature and can be expressed as [16]:

$$r_0(x) = \rho_0\big(1 + \beta T(x)\big) \qquad (1.3)$$

In the equation, ρ_0 is the unit length resistance of the metal at reference temperature, β is the temperature coefficient, and $T(x)$ is the thermal profile along the length of the interconnect. According to the distributed *RC* Elmore delay model, signal propagation delay through an interconnect of length L can be given as [17]:

$$D = R_d\left(C_L + \int_0^L c_0\,(x)dx\right) + \int_0^L r_0\,(x)\left(\int_x^L c_0\,(y)dy + C_L\right)dx \qquad (1.4)$$

In the equation, R_d is the output resistance of the driver, $r_0(x)$ and $c_0(x)$ are the resistance and capacitance per unit length at location x, respectively, and C_L is the load capacitance.

From the equations given in (1.2), (1.3) and (1.4), the interconnect delay model, which is dependent on the temperature of substrates, can be derived as [18]:

$$D = D_0 + (c_0 L + C_L)\rho_0 \beta \int_0^L T(x)dx - c_0\rho_0 \beta \int_0^L xT(x)dx \qquad (1.5)$$

, where

$$D_0 = R_d(c_0 L + C_L) + (c_0\rho_0 L^2/2 + C_L\rho_0 L) \qquad (1.6)$$

D_0 is the Elmore delay of an interconnect corresponding to the unit length resistance at 0℃. From equation (1.5), we can derive that there will be roughly five to six percent increase in the Elmore delay of a long global interconnect for each uniform temperature increase of 20℃ in its corresponding substrates [16]. Therefore, if we do not consider the thermal distribution of substrates in selecting optimal paths for global interconnects especially when they pass through hot regions or hotspots, then delay-induced timing failure or delay-related performance degradation might be unavoidable.

1.1.3. Power consumption

CMOS circuits are built on the basic structure of a pair of *n*-type and *p*-type MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors) in series. One transistor of the pair stays off except when it changes its states from on to off or vice versa. Thus, logic gates based on CMOS technologies consume negligibly small amount of power when not in switching in theory. However, aggressive technology scaling and rapid increase both in chip density and clock speed in recent years have caused dramatic increase in power consumption, and as a result, power consumption became a critical factor in chip design and a major issue in semiconductor industry.

Two major sources of power consumption in a chip are dynamic power and static power as we can see in equation (1.7):

$$P = \text{Dynamic power consumption} + \text{Static power consumption}$$
$$= \alpha C_L V^2 f + V I_{leakage} = \alpha C_L V^2 f + V\left(I_{subthreshold} + I_{gate_oxide}\right) \qquad (1.7)$$

P in (1.7) is the total power consumption of a device, and the first term is the dynamic power portion when we can ignore the power dissipation due to the direct path short circuit current when both NMOS and PMOS transistors are active for a short period of time during the gate voltage transition. In the equation, the average switching activity factor is denoted by α, which is roughly 0.2 for logic blocks in 65nm technology [19], and C_L is the total load capacitance of all gates in the device, and V and f are the supply voltage and the clock frequency, respectively.

Dynamic power consumption basically results from the repeated charging and discharging of load capacitance on the output of the gates, and it has been a principal source of the total power consumption for several technology generations as we can see in Figure 1.5. Dynamic power is proportional to the square of the supply voltage as we can see in equation (1.7), and CMOS technology scaling and the reduction in the supply voltage have managed dynamic power consumption relatively well, especially when compared with the explosive increase in static power consumption as given in Figure 1.5. However, dynamic power will still keep increasing in the future due to the continuous increase in power density or the number of gates in a chip and the limit to the supply voltage reduction as we can see both in Figure 1.6 and

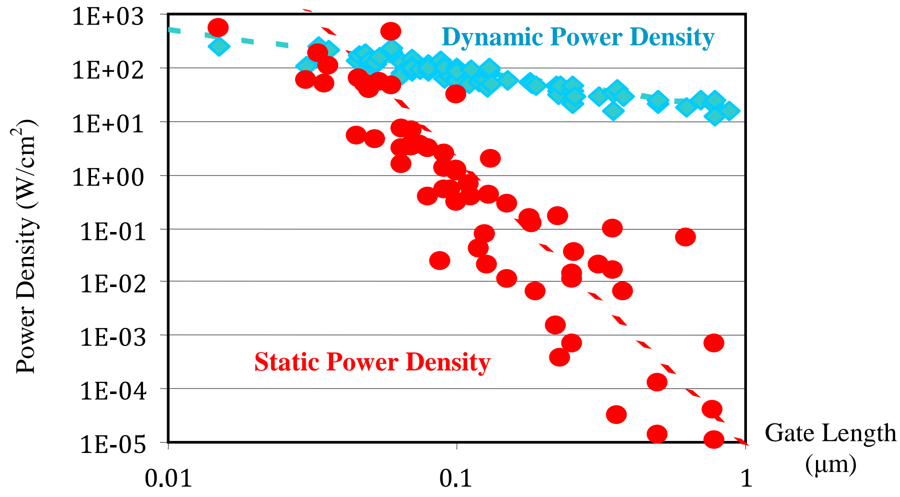


Figure 1.5. Trend in power density as a function of gate length [85]

Figure 1.7, which show the trend in power consumption of SoC (Systems on Chip) consumer portable chips and SoC consumer stationary chips, respectively.

Static power consumption arises from leakage current, which is the combination of subthreshold leakage current and gate-oxide leakage current. Subthreshold leakage is due to the current from source to drain when the gate voltage is smaller than the threshold voltage of a transistor, and it depends on the threshold voltage and the supply voltage of a transistor. Subthreshold leakage current can be modeled as [20]:

$$I_{subthreshold} = K_1 W e^{-V_{th}/nV_\theta}\left(1 - e^{-V/V_\theta}\right) \qquad (1.8)$$

In equation (1.8), K_1 and n are constants that are determined experimentally, W is the gate width or the width of a transistor, V is the supply voltage, V_θ and V_{th} are the thermal voltage and the threshold voltage, respectively. The thermal voltage V_θ is defined as kT/q, where k is

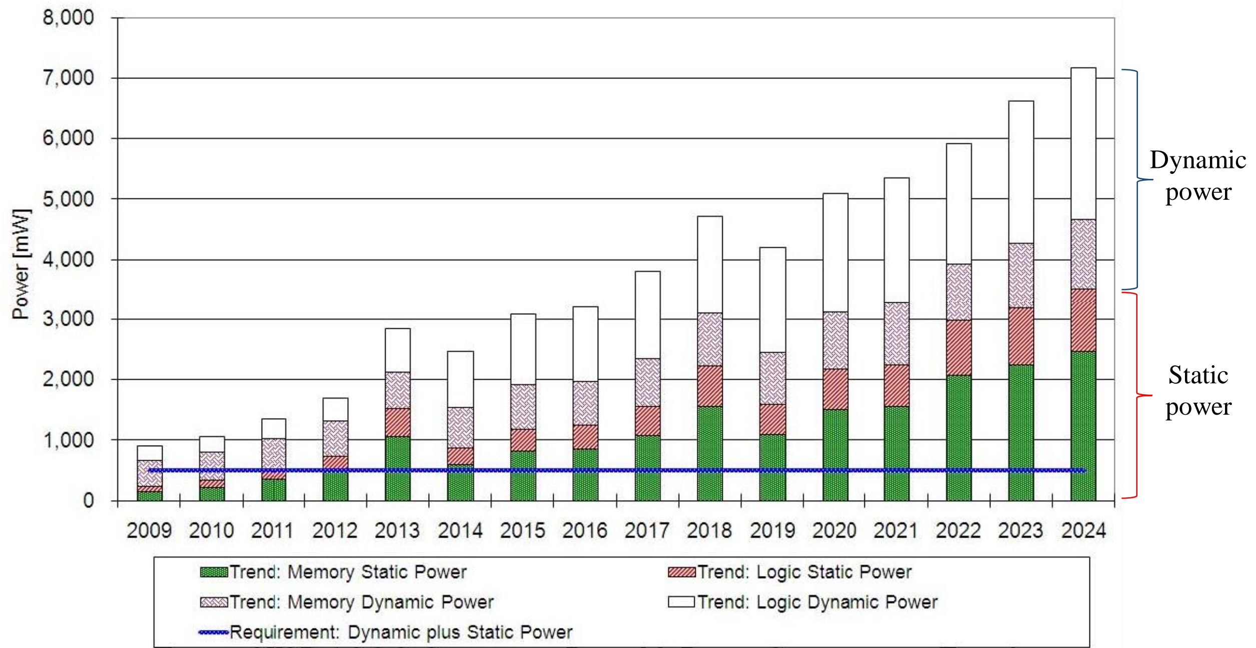


Figure 1.6. Trend in total chip power: portable consumer Systems on Chip (SoC) [88]

the Boltzmann's constant, T is the absolute temperature in Kelvin, and q is the elementary charge. It is roughly $26mV$ at room temperature, and as we can see in the definition, it increases linearly as temperature rises. The first thing that we can notice from the equation (1.8) is the exponential relationship between the subthreshold leakage current and the threshold voltage of a transistor. With technology scaling, the supply voltage of transistors has been consistently lowered as an efficient way of reducing the dynamic power due to the quadratic relationship between the supply voltage and the dynamic power consumption. The threshold voltage of transistors has been also getting lowered accordingly in order to maintain high switching speed because the delay of a transistor is inversely proportional to the difference between supply voltage and threshold voltage [21]. This incessant reduction in the threshold voltage of transistors brought about the exponential increase in the subthreshold leakage current and the increase in static power consumption as a result. The second one that

we can notice from the equation is that the subthreshold leakage current and static power consumption will increase exponentially when temperature rises owing to the linear relationship between temperature and V_θ. In extreme cases, thermal runaway can happen due to a positive feedback between the subthreshold leakage current and temperature; increase in subthreshold leakage current can lead to the rise of temperature, and the resulting high temperature will cause the increase in V_θ, and as a result, the increase in subthreshold leakage current in turn. Therefore, in order to minimize the power consumption, static power consumption to be exact, and also to prevent temperature-induced chip failures, it is crucial to manage the temperature of a chip efficiently.

Another source of leakage current is gate-oxide leakage, which is due to the tunneling current through the gate-oxide insulator, and it mainly depends on the supply voltage and

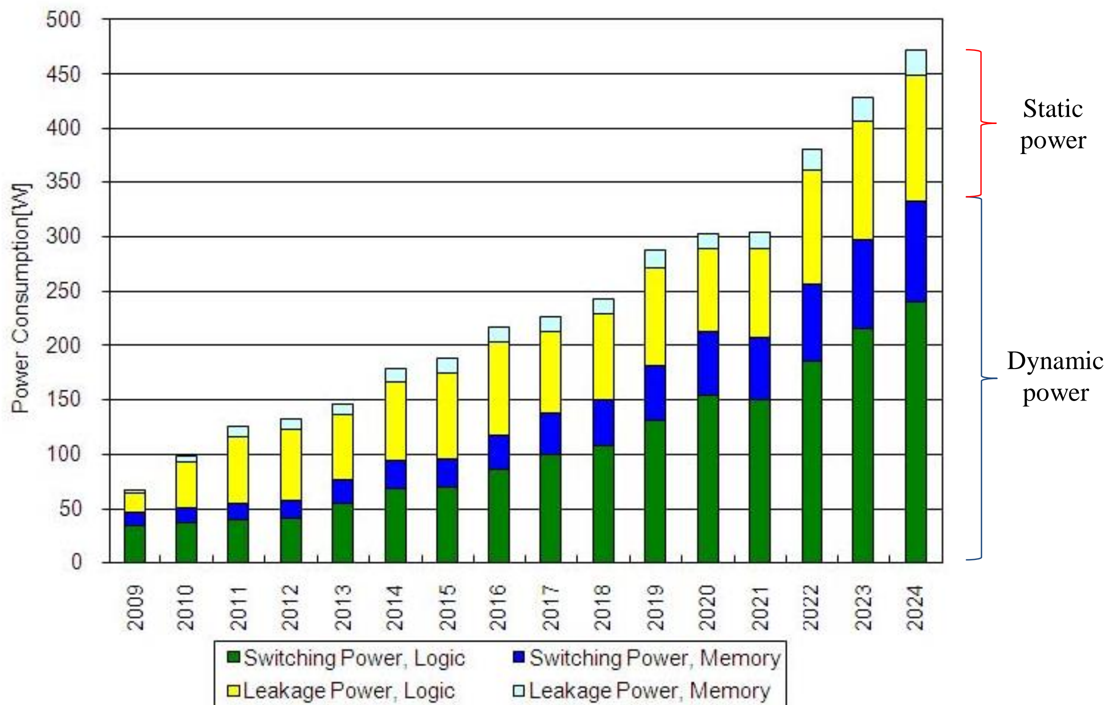


Figure 1.7. Trend in total chip power: stationary consumer Systems on Chip (SoC) [88]

oxide thickness. It was shown that the gate-oxide leakage current does not depend on temperature when compared with the subthreshold leakage current [22], and it can be roughly modeled as [20]:

$$I_{gate_oxide} = K_2 W \left(\frac{V}{T_{ox}}\right)^2 e^{-\alpha T_{ox}/V} \qquad (1.9)$$

In the equation, K_2 and α are constants that are determined experimentally, W is the gate width, V is the supply voltage, and T_{ox} is the thickness of gate dielectrics or gate oxide. We can see that the reduction in the supply voltage and the increase in the thickness of gate dielectrics will be helpful in minimizing the gate-oxide leakage current. Unfortunately, the thickness of gate dielectrics has been reduced continually with the technology scaling; thus, it became necessary to find a way of increasing the effective thickness of dielectrics in order to minimize the gate-oxide leakage current. As a solution to this issue, metal gates and high-*k* dielectrics came into play in current CMOS technology nodes instead of traditional polysilicon gate electrodes and silicon dioxide gate dielectrics [23].

1.2. Dynamic Thermal Management (DTM)

As we discussed in previous sections, temperature plays a critical role in the reliability, performance, and the power consumption of a chip in current and future CMOS technology nodes. Therefore, temperature of a chip, especially in case of a high performance chip, should be managed in a smart way at runtime so that the maximum temperature can be controlled and

also temperature can be evenly distributed both temporally and spatially for better reliability and performance of a chip. According to [24], cost for the implementation of cooling and packaging solutions was expected to increase at an alarming rate with the thermal dissipation of 65W or higher; thus, thermal management of a high performance chip is also quite crucial in terms of cooling and packaging cost. A large number of techniques for Dynamic Thermal Management (DTM) have been proposed and developed in recent years as ways of limiting the peak temperature of a chip or managing the temporal and spatial temperature variation of a chip through proper resource management [25] [26]. Those techniques can be roughly classified into one of two categories based on how the source management is performed: hardware-based techniques and software-based techniques.

1.2.1. Hardware-based DTM

The relationship between temperature and power consumption is quite complicated, but temperature can be managed to a certain extent by controlling power consumption of a chip. One of the simple power management techniques, which is called clock gating, began to be used widely in the early 2000's [24]; dynamic power consumption can be minimized by disabling the clocks in a functional block when the functional block is not in use or when the temperature of the functional block reaches a threshold. Clock gating is relatively simple to implement and has good cooling capability because we can effectively reduce the power consumption of a clock tree, which may consume up to around 70% of total dynamic power [27], but the performance degradation is quite high.

Changing the supply voltage and the clock frequency of a processor dynamically based on the workload can be effective in reducing the dynamic power consumption because of the quadratic relationship between dynamic power and the supply voltage, and this technique is called DVFS (Dynamic Voltage and Frequency Scaling) [28]. In case of a processor consisting of multiple cores, the supply voltage and the clock frequency settings of each core can be scaled independently, and it is termed local DVFS or distributed DVFS or per-core DVFS [29], while the chip-level voltage and frequency control is usually termed global DVFS [30]. Additional hardware components and increased design complexity to support multiple clock domains or multiple voltage/frequency islands (VFI) might become a critical issue especially in case of processors with a large number of cores [31].

Fetch gating [28] [32] is another way to cool down a chip through power consumption reduction; it controls the instruction activity in the pipeline by throttling the fetch stage, and its performance on power reduction and thermal management highly depends on the implemented throttling mechanisms as expected.

1.2.2. Software-based DTM

A simple temperature-aware task scheduling technique for single-threaded processors was proposed in [33]; kernel monitors the CPU activity of each process and the temperature readings from a thermal sensor. When the temperature of a chip becomes higher than a threshold, the kernel identifies processes that use more CPU activities than a predefined value, and then it slows them down for cooling purpose. Even though it was simple, it worked effectively to some extent. This basic idea was extended to temperature-aware scheduling

techniques for processors that support multithreading or have multiple cores. For example, a temperature-aware scheduling technique for Simultaneously Multithreading (SMT) processors was proposed in [34]; it manages the execution of threads selectively and dynamically based on the probability of heat generation of each thread, and hardware event counters [35] are used for the estimation of the heat generation probability. In [36], a scheduling method specifically targeting MPSoC (Multi-Processor Systems on Chip) was proposed; for each core or processor, the probability of workload assignment is calculated and updated regularly based on the temperature history in the past, and one core with the highest probability is selected when a new workload assignment is required.

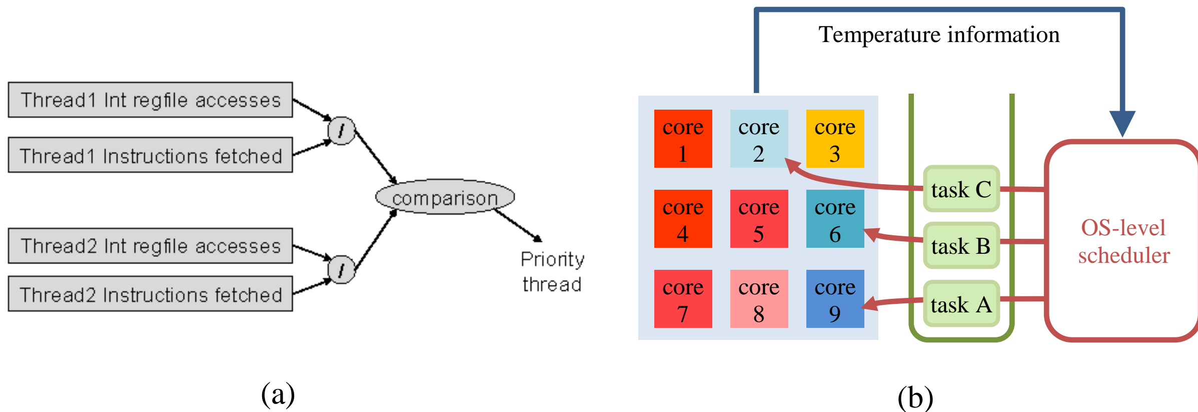


Figure 1.8. (a) Thread selection when the integer register file is thermally critical [34], (b) Temperature-aware task scheduling for MPSoC [36]

When there are multiple cores or processors in a chip, process or task migration can be used effectively in order to balance the thermal distribution among all cores and also to improve the performance as a result; in [2], a task migration technique was used on top of local DVFS, and it successfully avoided all thermal emergencies and also achieved 2.6 times speedup when compared with the base case of using local clock gating without task migration.

1.3. Thermal sensors

As we can expect naturally, DTM solutions use temperature information in order to manage the thermal distribution of a chip. Performance Counter-based temperature information can be used for thermal management [35], but the information is not a direct representation of thermal behaviors of a chip most of the time, and it just supplies approximation at best. In that sense, it is far better to use the temperature information from thermal sensors because it represents actual thermal behavior of a chip. Each thermal sensor basically provides point-wise temperature information. Thus, it would be better to use a large number of thermal sensors in order to have correct temperature information at any locations of interest on a chip. As for the locations of interest, hotspots need to be monitored first for better reliability and performance, and also for the reduction in power consumption of a chip just as we discussed in previous sections. In addition, a lot more thermal sensors need to be deployed across a die so that the thermal distribution over a die can be monitored and balanced out for the increased reliability of a chip and also for the prevention of performance degradation. However, it is not reasonable to allocate as many thermal sensors as possible on a small-sized chip in reality due to a lot of practical design constraints [37]: power consumption and heat generation of thermal sensors, routing and placement issues, etc. As a result, quite a large number of methods have been proposed regarding how to select the number of thermal sensors properly and how to allocate them on a die in order to have accurate temperature readings at any locations of interest on a die at runtime.

Another issue to be resolved is the accuracy of thermal sensors. A thermal sensor in a 0.35 μm 2.5V digital CMOS technology, which was implemented in a general purpose

microprocessor in the late 1990's, had the reading accuracy of $\pm 12℃$ with the resolution of $4℃$, and the area and the maximum power consumption of the sensor were 0.192mm^2 and 10mW, respectively [38]. Since then, great improvement has been made in its reading accuracy, size, and power consumption, but there still remains a lot of work to be done, especially when it comes to the design of on-chip thermal sensors that are fully compatible with digital CMOS technologies.

1.3.1. Sensor allocation: hotspot monitoring

In the late 1990's and the early 2000's, some general purpose microprocessors came with a single on-die thermal diode that could be used to monitor the temperature of a chip and also to trigger a signal to shut down the chip for protection purpose when the temperature of a chip reached a critical point [39]. Since then, thermal distribution of a chip has become a lot more complicated due to a large number of hotspots spreading across a die, and multiple thermal sensors came into play to monitor the temperatures of hotspots more efficiently and accurately.

a) Uniform allocation

One simple way to place multiple thermal sensors on a die will be to put them on a uniform grid, and it is not necessary to consider the thermal distribution of a chip with this allocation method. As a result, some hotspots might not be detected, and the accuracy will be quite limited especially when a small number of thermal sensors need to cover the entire die area and when the spatial thermal variation of a chip is quite large. In order to overcome this shortcoming, a linear interpolation technique using the temperature readings of four neighboring thermal sensors was proposed for the estimation of the maximum temperature of

a chip [40]; a single-core processor and SPEC2000 benchmark suites [41] were used for the simulation, and it achieved the maximum error of 5.47℃ and the averaged error of 1.05℃ using 16 thermal sensors on a 4 by 4 uniform grid.

b) Recursive bisection

When thermal distribution of a chip is available, this information can be used for sensor allocation, and thermal sensors can be allocated in a smart way so that the hotspots of a chip can be monitored correctly while minimizing the number of thermal sensors. Initial emphasis was placed on how to group multiple hotspots efficiently into a small number of clusters so that a single thermal sensor can monitor all hotspots in each cluster properly. In [42], a sensor allocation method based on recursive bisection was proposed; this algorithm divides the die area into an array of blocks using the information about hotspot locations, and the size of each block is adjusted in such a way that all hotspots in each block can be covered and monitored by a single thermal sensor assigned to the block. Experimental results showed that the number

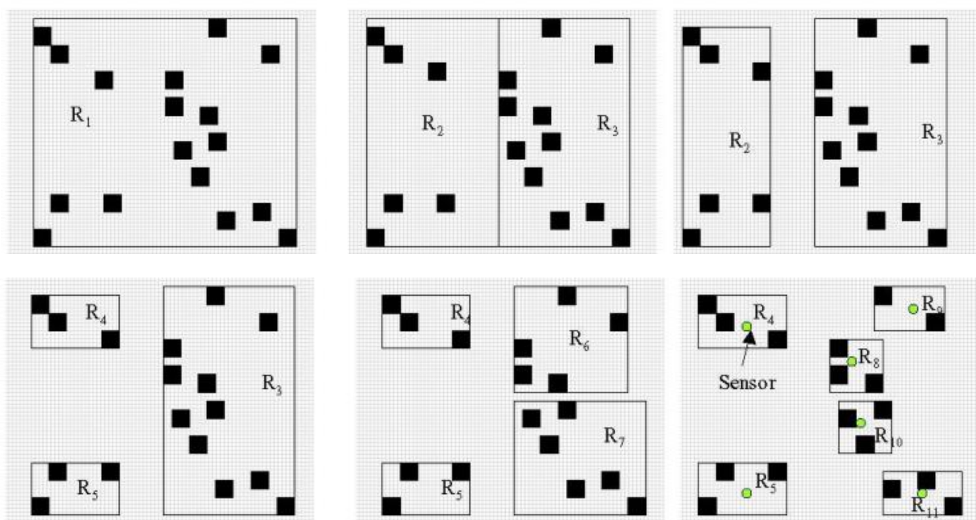


Figure 1.9. Recursive bisection based thermal sensor allocation [42]

of required thermal sensors was 19 for an FPGA with 96 by 64 CLBs (configurable logic blocks), while 40 thermal sensors were required in case of grid-based uniform allocation. This method works well when the number of hotspots is not large, but with the increase in the number of hotspots, a lot more thermal sensors will be required.

c) Temperature-aware *k*-means clustering

A thermal sensor allocation method based on k-means clustering [43] was proposed in [44]; each and every hotspot is assigned to one of k clusters recursively, where k is the number of thermal sensors, so that the Euclidean distance between the centroid of a selected cluster and the hotspot is minimized. Then, k thermal sensors are assigned to the centroids of those k clusters, and each thermal sensor represents the thermal status of each cluster. One modification made in [44] to the basic k-means clustering method is that the temperature difference between the centroid of a cluster and a hotspot is included as a third dimension in Euclidean distance calculation, as given in (1.10):

$$E(O_j, h_i) = (O_{jx} - h_{ix})^2 + (O_{jy} - h_{iy})^2 + (O_{jt} - h_{it})^2 \qquad (1.10)$$

In the equation, (O_{jx}, O_{jy}) represents the x and y coordinates of the centroid of jth cluster out of k clusters, and (h_{ix}, h_{iy}) represents the x and y coordinates of ith hotspot. The temperature values at the centroid of jth cluster and i^{th} hotspot are represented by O_{jt} and h_{it}, respectively. That is, the main idea was to consider temperatures in the process of hotspot clustering, and the results based on a single-core processor and SPEC2000 benchmark suites [41] were reasonable with the maximum error of 6.11℃ and the averaged error of 2.66℃

using 16 thermal sensors. However, this method might produce some unreasonable results, especially when remotely located hotspots have smaller temperature differences than closely located hotspots.

1.3.2. Sensor allocation: full-chip profile reconstruction

In recent years, a large number of new sensor allocation methods have been proposed to support full-chip thermal profile reconstruction at runtime from the temperature readings of a small number of sensors. Sensor allocation is performed with a view to a better runtime thermal profile reconstruction from the beginning, and the number and the locations of thermal sensors are determined accordingly. Fine-grain DTM solutions can make full use of the detailed temperature information from full-chip profile reconstruction, especially on multi-core processors [45]; task migration among cores can be performed more efficiently, and the thermal behavior and static power consumption of caches, which consume a large portion of the die area, can be optimized [46] [47].

a) Energy-aware distribution

In [48], energy analysis in frequency domain was used for sensor allocation. The main idea of this method is that the large thermal variations in thermal profiles correspond to large amount of energy in high frequency components in frequency domain. That is, thermal sensors should be distributed in proportion to the high frequency energy in frequency domain so that a lot more thermal sensors can be assigned to regions with large thermal variations. This method alternates vertical bisection and horizontal bisection, and compares the high frequency energy of two bisected regions. Thermal sensors are allocated proportionately, and

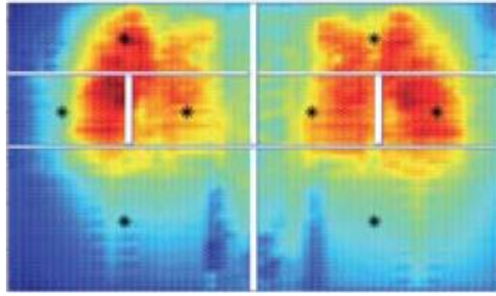


Figure 1.10. Energy-aware thermal sensor allocation [48]

the bisection continues until all thermal sensors are assigned. As for the full-chip thermal profile reconstruction, Discrete-Cosine Transform (DCT) [49] is used for the transformation between spatial domain and frequency domain; the temperature readings of thermal sensors in spatial domain are first transformed to the coefficients in frequency domain, and then the coefficients of low frequency components are transformed back to spatial domain to build full-chip thermal profiles. According to the experimental results on a dual-core processor and SPEC2006 benchmark suites [50] [51], the hotspot temperature error was around 16% to 21% of the temperature difference between the maximum and the minimum temperature of the chip, and the averaged absolute error over a die was around 14% with the use of four thermal sensors. This method considers only the amount of energy in AC or high frequency components in its sensor assignment, excluding DC component. As a result, it is possible to allocate a large number of thermal sensors in regions where no hotspots exist at all and allocate just a small number of sensors near hotspots. Considering the fact that it was proposed with full-chip thermal profile reconstruction in mind, this approach might have been unavoidable. However, hotspot monitoring might become an issue when just a small number of sensors are used for a chip whose thermal distribution is quite complicated. In addition to the hotspot monitoring issue, another possible issue we can surmise from the Nyquist

sampling theorem [52] is that quite a large number of thermal sensors need to be used in order to maintain good results when the thermal distribution of a chip has a lot of non-negligible high spatial frequency components.

b) Statistical approach

In [53], a statistical methodology was developed for sensor allocation and full-chip thermal profile reconstruction; the entire die area was divided into a 16 by 16 grid, and a set of nodes on the grid were selected so that the thermal correlation among them can be minimized, and the thermal correlation between the selected nodes in the set and the nodes outside the set can be maximized at the same time. In this way, each thermal sensor can provide as much temperature information as possible on the non-sensor nodes, while the redundancy among the sensor nodes is minimized. As for the full-chip profile reconstruction, temperatures at non-sensor nodes can be estimated based on the statistical correlation between thermal sensors and those non-sensor nodes. If we can allocate thermal sensors so that the correlation between sensor nodes and non-sensor nodes can be always high, then we can expect good results from this approach, but it doesn't guarantee accurate profile reconstruction always by its nature. Experimental results on SPEC2000 benchmark suites [41] and a processor, of which power consumption was rated 60W, showed that the Root Mean Squared Error (RMSE) was around 10℃ when four thermal sensors were allocated on a 16 by 16 grid.

c) Sensor-assisted power estimation

One way to have accurate temperature information of a chip is to solve the heat differential equation directly with correct power information. Performance counter-based

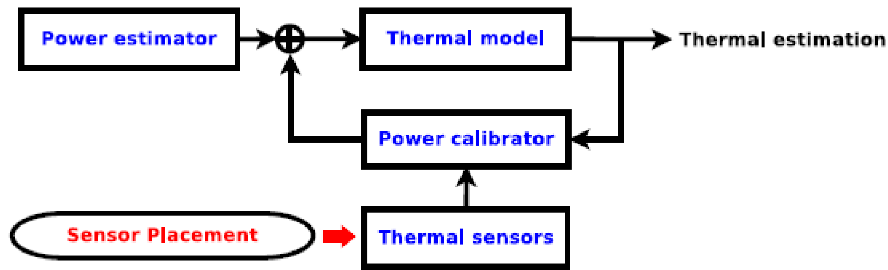


Figure 1.11. Thermal profile estimation based on sensor-assisted power estimation [56]

runtime power estimators [54] [55] can be used to supply power information at runtime, but they tend to have some power estimation errors. A new approach to achieve good temperature estimation based on the differential equation was proposed in [56], and it exploits the temperature readings of thermal sensors to correct the power estimation errors. In order to reduce the number of thermal sensors, functional blocks on a chip are clustered into sensor blocks based on the correlation in power estimation errors, and then a thermal sensor is assigned to each sensor block. According to the simulation results on a dual-core processor and SPEC2000 benchmark suites [41], it achieved the maximum error of 1.2℃ and the averaged error of 0.085℃ with six thermal sensors. The simulation results look promising, but one possible issue of this method is it depends on statistical information, i.e., correlation among multiple functional blocks, in correcting power information of multiple functional blocks. Therefore, its performance could be degraded for some applications where high correlation among functional blocks cannot be expected.

1.4. Dissertation overview

Issues regarding reliability, performance, and power have made DTM solutions play a vital role in deep submicron eras. Considering the fact that DTM solutions take actions based on the temperature information of a chip, the accuracy of the information is quite critical. BJT (Biploar Junction Transistor) based on-chip thermal sensors have long been used mainly owing to their excellent reading accuracy, while their size and power consumption leave a lot to be desired especially in current CMOS technology nodes. As an alternative, Ring-Oscillator (RO) based thermal sensors have been drawing a huge amount of attention lately due to their small size, low power consumption, and full compatibility with current digital CMOS technologies. One issue to be thoroughly considered when using RO-based thermal sensors is their limited accuracy of around ±3℃ [57]. The first proposal in this dissertation is regarding how to improve the reading accuracy of this type of on-chip thermal sensors. We propose the use of multiple virtual sensors at one location, and we make improvements by adaptively selecting one out of multiple virtual sensors using the temperature readings in the past and the calibration points of those multiple virtual sensors. In the case where we used two virtual thermal sensors, maximum absolute error was reduced down to 0.84℃ in comparison with 2.43℃ of a single physical sensor case, and the RMSE was reduced by 59%.

DTM solutions generally make use of point-wise temperature information from multiple thermal sensors. When we allocate those multiple thermal sensors on a die, it is reasonable to limit the number of thermal sensors because of area, power, and routing issues accompanying the sensor allocation. Therefore, it is crucial to choose the proper number of sensors and their locations on a die so that DTM solutions can manage the resources in an efficient way. Fine-

grain DTM solutions might be able to work more effectively when temperature information on a full-chip scale is available, especially when the thermal distribution of a chip is complicated. Our second proposal is intended to address these issues: how many thermal sensors are to be deployed, where to place them on a die, and how to reconstruct thermal profiles on a full-chip scale. A geometrical framework based on thermal gradient analysis was established, and this framework was used to choose the number of thermal sensors and their locations on a die. Thermal profiles on a full-chip scale were also reconstructed based on the framework and the temperature readings from the allocated sensors. Using six thermal sensors on a dual-core microprocessor, we achieved a 36% reduction in RMSE and a 50% reduction in averaged absolute error in comparison with a similar full-chip thermal profile reconstruction method given in [48].

In our third proposal, we discuss a physical design issue to improve the reliability and also to maximize the performance of a chip. The life expectancy and the amount of signal delay of global interconnects are dependent on the thermal distribution of substrates, and we can alleviate those issues by routing them in a smart way. The reduction in life expectancy can be minimized by routing them so that hotspots can be avoided as much as possible. Temperature-dependent delay can be also minimized by considering the thermal distribution of each possible path. Regarding the number of segments that pass through hotspots on a die, our proposed method reduced it by 42% on the average in comparison with a conventional router and also reduced it by 13% on the average when compared with a similar temperature-aware router [58]. When it comes to signal delay, our method achieved a 3.5% reduction on the average, while other temperature-aware router [58] increased the delay in some benchmark circuit cases.

1.5. Application to a chip design

As we briefly discussed in the previous section, we investigate in this dissertation how to manage the reading accuracy of on-chip thermal sensors, where to allocate those thermal sensors on a die, and how to route global interconnects using temperature gradient analysis. All these proposals can be applied to a chip design from its early stage, and possible

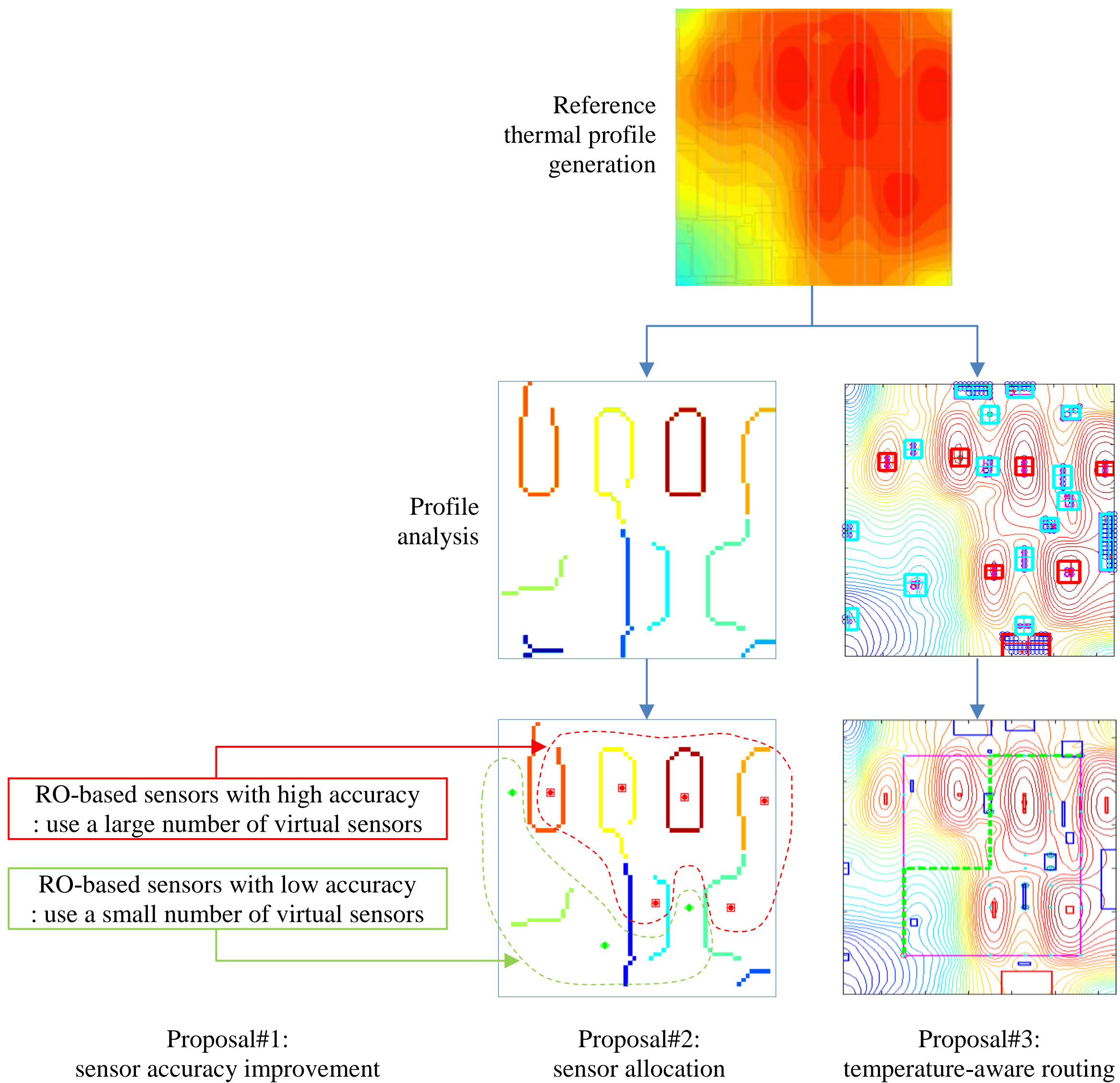


Figure 1.12. Application of propose methods to a chip design

application scenarios are outlined in this section.

At the early stage of chip design, the worst case power consumption of each and every functional block is estimated, and then thermal simulations are performed based on the floorplanning result of all functional blocks on a die in order to check the maximum temperature of a chip and also to initiate the design of cooling and packaging solutions. Thermal profiles from the simulations can be used for thermal sensor allocation and interconnect routing, and our proposed methods can be applied for that purpose. When a test chip is available, we can measure or capture detailed thermal profiles of a chip while running benchmark suites or test applications on it. However, at the early stage of a chip design, it is difficult to use a test chip for data collection or to estimate those thermal profiles accurately through simulations. With this limitation in mind, we use two different data sets in our sensor allocation proposal, which is presented in chapter 3; one data set is composed of thermal profiles of a test chip captured by a thermal imaging device, and another data set is composed of profiles generated by a thermal simulator based on the worst case power consumption estimation. When using thermal profiles from the worst case power consumption information, it is possible to miss a few local hotspots on a die, but experimental results show that we can still achieve good results using those thermal profiles.

Our sensor allocation method places thermal sensors on a die automatically using the analysis of the thermal profiles of a chip, and it allows designers to choose the total number of thermal sensors flexibly based on the availability of resources such as white space on a die. When necessary, our proposed method can be easily modified to choose the total number of sensors in a fully automated way based on several criteria. Firstly, die area and power

consumption associated with new thermal sensors and their interconnects can be used to set the maximum number of thermal sensors to be deployed. Secondly, the number of sensors can be increased incrementally until the improvement in error reduction falls below a pre-defined threshold. In this way, we can find the optimal number of thermal sensors with respect to the design objectives without violating major design constraints.

When it comes to the sensor allocation for a multi-core chip with hundreds of cores on a die, it might be unnecessary or even inefficient to perform full-chip scale analysis and sensor allocation, especially when an identical basic building block is uniformly distributed and placed over a die. Due to the inherent independence among the cores and the lack of structure and predicability in task assignment among hundreds of cores, it becomes difficult and meaningless to find a full-chip thermal profile that can represent all possible thermal profiles on a die for the purpose of analysis. In that case, it would be more effective to analyze the thermal characteristics of the basic building block and replicate the sensor allocation result on the building block to all the other blocks on a die.

Thermal sensors allocated by our proposed method are classified into two groups automatically: sensors that are mainly used to monitor hotspots and cold spots on a die and sensors that are mainly used for the improvement in thermal profile reconstruction. Therefore, we can use thermal sensors with high accuracy for the ones belonging to the former group and use sensors with low accuracy for the latter group when thermal sensors with high accuracy cannot be arranged for all sensors due to some resource issues. As for the accuracy of thermal sensors, BJT-based thermal sensors can be used at the locations where high accuracy is required, while RO-based sensors can be used at the remaining sensor locations. If we use

RO-based thermal sensors at all sensor locations and control the accuracy using the second proposal in this dissertation, then we can flexibly choose the number of virtual sensors depending on the required accuracy at each sensor location. Considering the high cost for sensor calibration, it is preferable to minimize the number of calibration points or the number of virtual sensors. In that sense, it would be a reasonable choice to use two virtual sensors for most cases, with which we still achieve the maximum absolute error of 0.84℃ with two-point calibration, and to use a larger number of virtual sensors for some limited cases only.

Thermal profiles and their analysis results used in sensor allocation can also be used in routing global interconnects. Using the proposed method, we can improve the reliability of interconnects and also minimize temperature-induced signal propagation delay of the interconnects at the same time. It also means we can push the temperature of a chip somewhat further in order to achieve better performance, not worrying about the reliability of a chip.

In the following chapters, each proposal will be presented in further detail: improvement in the reading accuracy of RO-based on-chip thermal sensors in chapter 2, thermal sensor allocation and full-chip thermal profile reconstruction in chapter 3, and temperature-aware routing for global interconnects in chapter 4.

CHAPTER 2. On-Chip Temperature Estimation using Multiple Virtual Sensors

2.1. Motivation

DTM solutions need on-chip thermal sensors to get the accurate temperature information of a chip at different locations at runtime. Ideally, each on-chip thermal sensor should consume negligible area and power, while instantly providing accurate temperature readings so that DTM solutions can work just as intended. In reality, each thermal sensor requires non-negligible footprint and also consumes a certain amount of power even though both quantities might be small. More importantly, temperature readings of thermal sensors will not be free from errors. According to recent studies, incorrect temperature readings could impact on a processor's power and performance indeed; the performance of DTM solutions could be improved by 14.3% when the RMSE in sensor readings was reduced by 71.5% since DTM actions could be triggered more appropriately with more accurate temperature readings [59]. Similarly, 1.5℃ accuracy in sensor readings was equivalent to 1Watt of processor power in mobile computing environment, and 1℃ accuracy was equivalent to 2Watts in the desktop computing environment [60].

A lot of researchers have been designing new on-chip thermal sensors that are small and consume little energy without compromising the power grid integrity of a chip; BJT-based thermal sensors generally show good performance relative to their accuracy [61] [62].

However, they are not fully suitable for digital CMOS technologies; RO-based thermal sensors are digital CMOS compatible and have low area and consume only nano-Joules of energy for each measurement (e.g., 0.2nJ per sample [63] [57]). However, RO-based thermal sensors suffer from low levels of accuracy (~±3℃ [57]), motivating the need for accuracy improvement.

In this chapter, we propose an efficient method to increase the accuracy of temperature readings through the use of multiple virtual thermal sensors that are generated from a small, low power, inaccurate physical thermal sensor (e.g., a CMOS RO-based thermal sensor) by adaptively switching its calibration points on the run.

2.2. Related work

Existing efforts take different approaches for accurate temperature estimation based on noisy temperature readings; statistical approaches [59] [64] use power and thermal simulation statistics extracted from benchmark suite execution to estimate correct temperatures and improve the accuracy by using the correlation between multiple sensors, which are assumed to be jointly Gaussian and highly correlated, at different locations; performance counter-based methods [65] combine noisy sensor readings with temperature estimations based on system performance counters; Kalman Filter based methods [66] improve sensor accuracy by combining noisy sensor readings with power consumption information of each functional block.

Statistical methods need prior simulation results and space to store the information and don't guarantee accurate estimation results either. Performance counter-based approaches

require power consumption data of each functional block, and the Kalman filter based methods may result in high computational complexity. In contrast, our approach, which uses multiple virtual thermal sensors generated from each physical thermal sensor, is much simpler while achieving good accuracy.

2.3. Cooperative temperature estimation

2.3.1. Modeling

RO-based on-chip thermal sensors are used in this proposal since they are fully compatible with digital CMOS technologies while consuming small area and power.

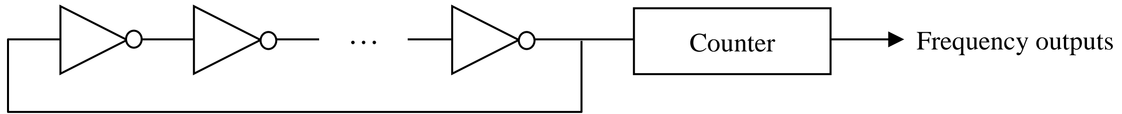


Figure 2.1. A RO-based on-chip thermal sensor

Figure 2.1 shows the basic structure of a RO-based on-chip thermal sensor that has an odd number of inverters. The transition time for each inverter to switch levels (H-L or L-H) is determined by several factors [59]. The high-to-low time is:

$$t_{PHL} = \frac{C}{\mu_n C_{ox}(W/L)_n(V_{DD} - V_t)}\left[\frac{2V_t}{V_{DD} - V_t} + ln\left(\frac{3V_{DD} - 4V_t}{V_{DD}}\right)\right] \qquad (2.1)$$

In the equation, μ_n is the mobility of electron in silicon, C_{ox} is the capacitance per unit gate area, $(W/L)_n$ is the ratio between width and length, V_{DD} is the supply voltage, V_t is the threshold voltage, and C is the load capacitance that the inverter drives. The low-to-high transition time (t_{PLH}) uses the same equation with PMOS parameters: μ_p, the mobility of hole, and $(W/L)_p$ instead.

$$f = \frac{1}{N(t_{PHL} + t_{PLH})} \qquad (2.2)$$

Equation (2.2) describes the frequency output of a thermal sensor [59], where N is the number of inverters in the sensor.

Due to process variation and environmental uncertainties, most of the parameters in (2.1) will be random variables, and it is reasonable to assume they follow a Gaussian distribution [64]. In addition, $\mu_{n(p)}$ and V_t are temperature-sensitive parameters, and we assume they follow these empirical equations, respectively [67] [68]:

$$\mu_{n(p)} = \mu_{n0(p0)}(T/T_0)^{-1.5} \qquad (2.3)$$

$$V_t = V_{t0} - 0.002(T - T_0) \qquad (2.4)$$

In above equations, T_0 is the room temperature, $\mu_{n0(p0)}$ is the nominal value of mobility, and V_{t0} is the threshold voltage at T_0.

2.3.2. Calibration

Due to process variation and environmental uncertainties, the frequency outputs of RO-based thermal sensors will be different from sensor to sensor; Figure 2.2 shows simulation-based variations in the frequency outputs of a number of sensors with exactly identical design parameters, and a reference frequency output, which is calculated from nominal parameter values, is also shown as a thick dashed line in the same figure for comparison purpose. We can observe the frequency output of each sensor has a different gain (or slope) and a different offset. Therefore, frequency outputs of each sensor should be interpreted or calibrated properly. That is, the conversion from the frequency outputs of a sensor to temperature readings should be performed with great care in order to minimize possible temperature reading errors [69]; single-point and two-point calibration are most popular, while for higher precision, piece-wise linear calibration or a nonlinear least square regression method based on multiple calibration points might be considered. Due to the fact that the actual frequency

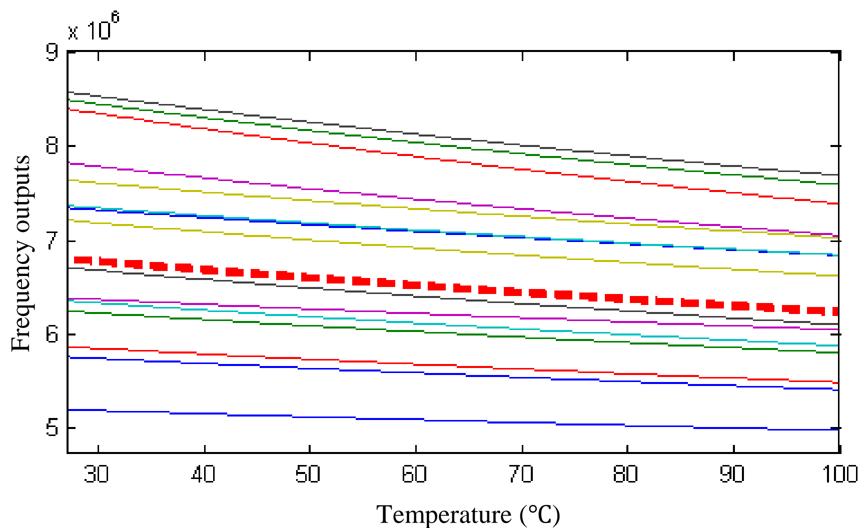


Figure 2.2. Variations in frequency outputs of RO-based thermal sensors due to process variation and environmental uncertainties

outputs of RO-based thermal sensors are a lot noisier than the simulation results shown in Figure 2.2, and do not always decrease or increase monotonically, piece-wise linear calibration or a nonlinear least square regression method using multiple calibration points might not give better results or drastic improvements over traditional single-point or two-point calibration in case of RO-based thermal sensors. Therefore, we adopt much simpler single-point and two-point calibration in this proposal.

In case of single-point calibration, either a gain or an offset is adjusted to generate correct temperature readings [69]. In other words, the frequency output at the single calibration point (temperature) is used as a reference point in the conversion from the frequency outputs of a sensor to the temperature readings. Two-point calibration usually gives better results than single-point calibration does because both a gain and an offset are adjusted together based on the frequency outputs of a sensor at two calibration points. However, if the gain or offset

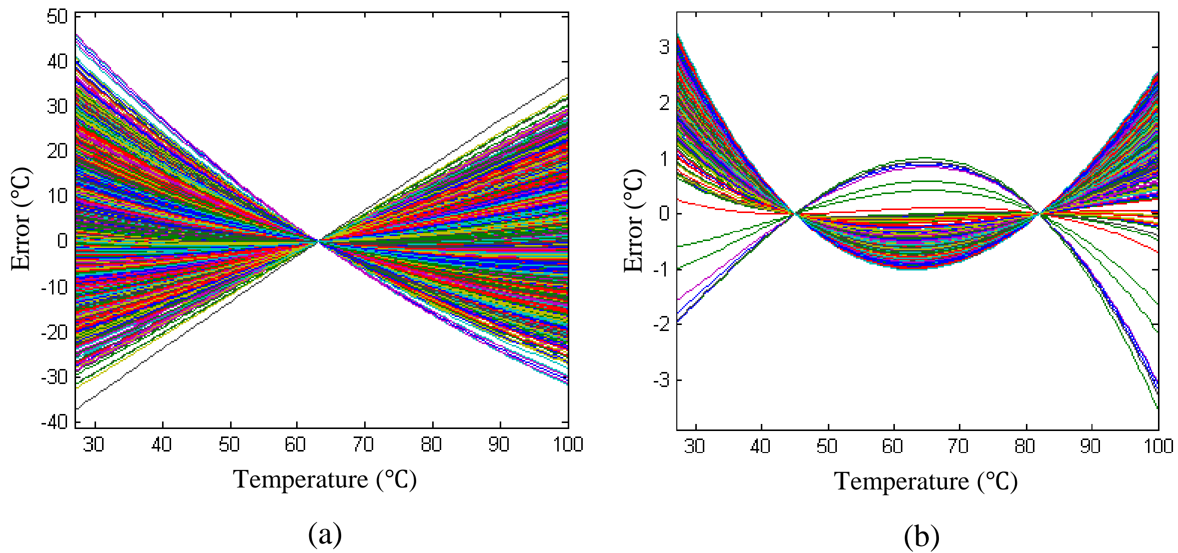


Figure 2.3. Temperature reading errors of 1,000 sensors due to variations: (a) with single-point calibration at 63℃, and (b) with two-point calibration at 45℃ and 82℃

varies a lot due to environmental uncertainties, neither single-point nor two-point calibration might give satisfactory results as shown in Figure 2.3, where temperature reading errors in ℃ are plotted as a function of temperature for 1,000 sensors with identical design parameters. To address this issue, we present a cooperative estimation strategy in order to improve the temperature reading accuracy.

2.3.3. Cooperative estimation

a) A sensor group

Temperature reading errors of a thermal sensor tend to become very small around each calibration point. We exploit this observation to achieve good error performance over an entire temperature range by using a sensor group composed of multiple physical sensors – each calibrated differently, but all at one sensor location – and judiciously select one of those

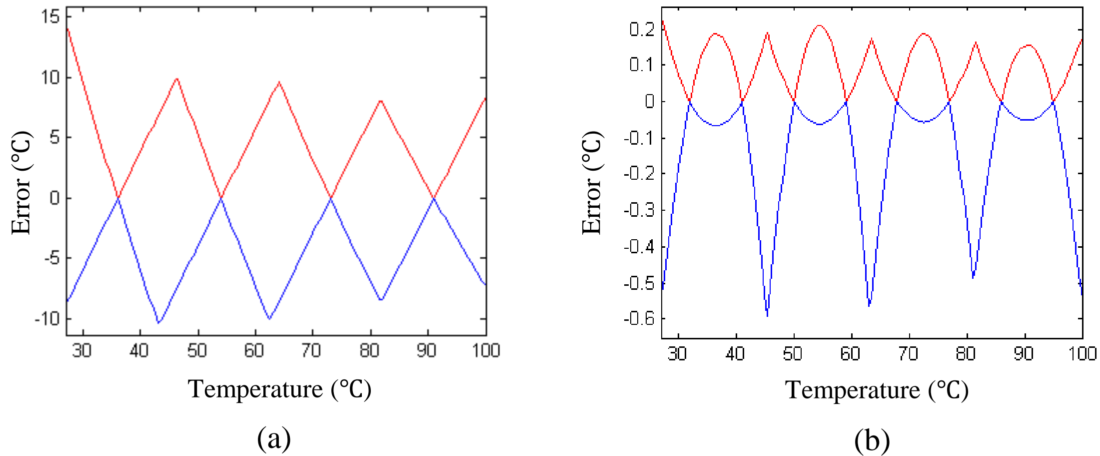


Figure 2.4. Ideal error bounds in case of four sensors at one location: (a) single-point calibration at 36, 54, 73, 91°C, respectively, (b) two-point calibration at (32, 41), (50, 59), (68, 77), (86, 95)°C, respectively

physical sensors on the run for the temperature readings at the location so that the errors over the entire temperature range are to be contained quite small. Figure 2.4 shows some examples of such error bounds for a sensor group composed of four physical thermal sensors, with single-point calibration on the left, and two-point calibration on the right.

b) Multiple virtual thermal sensors

While using a group of multiple physical thermal sensors with different calibration points at one location may provide immunity to failures, the area and power consumption overheads may be quite high. Instead, we can simulate multiple sensors from a single physical sensor, avoiding the overhead of multiple physical sensors at one location. Simply put, calibration is a process of interpreting the frequency outputs of a physical thermal sensor to generate correct temperature readings at the sensor location. Thus, by changing the interpretation, we can simulate multiple sensors at the location from one physical sensor; in case of single-point calibration, we can simulate multiple sensors, each of which has different calibration point, by simply switching one constant, either a gain or an offset; in case of two-point calibration, we need to switch two constants, both a gain and an offset, for the simulation of multiple sensors with different calibration points. Since the thermal profile of a chip usually stays constant for over a few hundred milliseconds [70], we can repeat the interpretation processes multiple times simulating thermal sensors with different calibration points as long as temperature does not change.

c) Switching virtual sensors

We can take several simple approaches in selecting one sensor out of multiple virtual ones in order to determine the correct temperature reading at time index i. For instance, variations

in the standard deviation of each virtual sensor in the past are traced over a predefined tap size *m*; then compared with each other; finally, one sensor with the smallest variation can be chosen to give the temperature reading at time index *i*. A much simpler approach would be to use the mean of temperature readings over a predefined tap size *n* in the past; this mean is compared with the calibration point of each sensor, and a sensor whose calibration point is the nearest the mean is selected; in case of two-point calibration, the center of the two calibration points of each sensor can be used to represent the calibration point of the sensor. We used the latter in our experiments with *n* set to three.

d) Prediction

Various prediction methods can be used together with the sensor switching method described earlier to improve estimation results. The simplest is a linear prediction method using the temperature readings over the last *k* taps in the past. A weight vector can be used to give more priority to more recent readings, as summarized in (2.5), with *k* set to four in the experiments. In (2.5), $T(i)$ is the temperature at time index *i*, and w_s is weight.

$$T(i) = T(i-1) + \frac{1}{k-1}\sum_{s=1}^{k-1} w_s \frac{T(i-1) - T(i-1-s)}{s} \qquad (2.5)$$

In the experiments, final temperature estimation at each time instance was made based on the temperature reading of a selected sensor and the prediction result with the ratio of 6 to 4.

2.4. Experimental results

We evaluated 1,000 RO-based thermal sensors using Monte Carlo simulations. For the parameters in (2.1), we assumed a Gaussian distribution with means and standard deviations as shown in Table I. 130nm CMOS technology process parameters were used for the simulations, and each value was set according to the predictions given in [64] [71] [72].

Our experiments used 20 thermal profiles that were generated randomly, summing up multiple sinusoids with different magnitude and phase terms, in order to simulate actual temperature variations of a chip.

For each possible combination of a calibration method (single-point or two-point calibration) and the number of sensors (one to four) at one location, we generated the upper and the lower error bound as a function of temperature using Monte Carlo simulations, and examples of these error bounds are shown in Figure 2.4. The generated error bounds were then used to randomize temperature readings of each sensor for a given thermal profile as a function of time, or to simulate actual temperature readings of each sensor by adding some noise to the thermal profile. For the randomization of temperature reading at each time

Table I. Random variables for Monte Carlo simulations

Parameters	$W_{n(p)}$ (nm)	$L_{n(p)}$ (nm)	T_{ox} (nm)	V_{DD} (V)	V_t (V)	$\mu_{n(p)}$ $(m^2/V \cdot s)$
Mean	100	49	2.25	1.3	0.288	0.03
Std. dev.	5%	6%	3%	5%	4%	3%

instance, we applied uniform randomization in order to maximize randomness, choosing an arbitrary value in a uniform manner between the upper bound and the lower bound at the corresponding temperature value.

Figure 2.5(a) shows the temperature readings of four sensors, each calibrated with single-point calibration at 36°C, 54°C, 73°C, and 91°C, respectively. Similarly, Figure 2.5(b) shows the temperature readings of four sensors, each calibrated with two-point calibration at (32°C, 41°C), (50°C, 59°C), (68°C, 77°C), and (86°C, 95°C), respectively. Horizontal lines in each subplot represent the corresponding calibration points of each sensor, and an original thermal profile to be estimated is also plotted as a black solid line in each subplot for reference purpose.

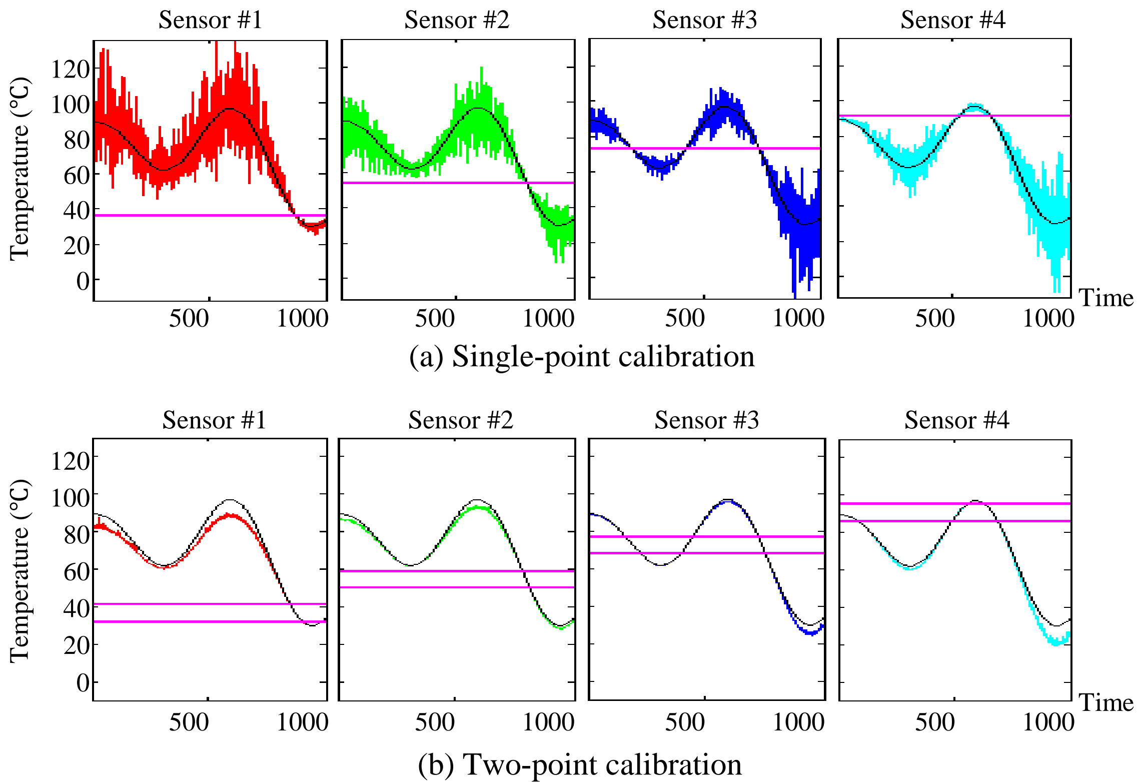


Figure 2.5. Temperature readings of four sensors: (a) with single-point calibration, (b) with two-point calibration

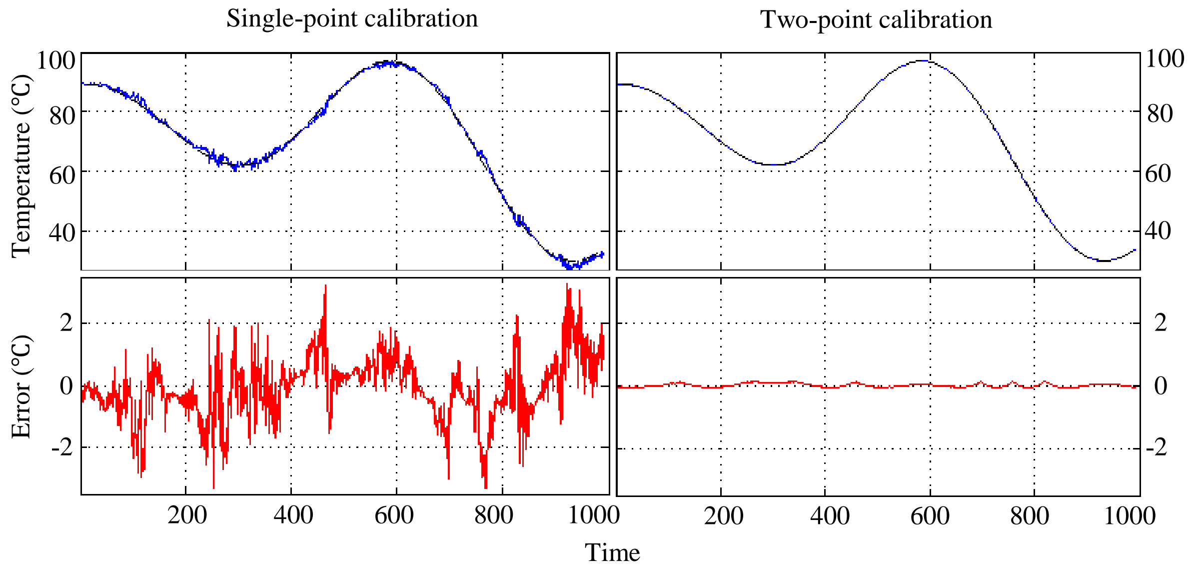


Figure 2.6. Estimation results (top) and corresponding errors (bottom) in case of four sensors: single-point calibration (left), and two-point calibration (right)

Estimation results and their corresponding estimation errors for one thermal profile are given in Figure 2.6; results based on four sensors with single-point calibration are given on the left, and results based on four sensors with two-point calibration are given on the right. From the figures on the right, we observe that the errors are almost negligible except for some time instances when the sensors are switched.

Maximum absolute errors and the averaged results based on 20 randomly-generated thermal profiles are summarized in Table II, Figure 2.7, and Figure 2.8. In Table II, we can observe that both maximum absolute errors and RMSEs decreased with the increase in the number of sensors and calibration points. RMSE reduction rates in Figure 2.8(b) are relative values compared with the RMSEs of a single sensor case, and we can observe the error reduction rate increased from 60.4% to 91.1% when we increased the number of sensors from two to four with two-point calibration. In case of single-point calibration, the error reduction

Table II. Maximum absolute error and RMSE of estimation for 20 profiles

Calibration method/ number of sensors		Max. abs. error in ℃	RMSE in ℃ (w/o prediction)	RMSE in ℃ (w/ prediction)
Single-point calibration	1	27.23	**2.93**	-
	2	8.01	2.90	**1.54**
	3	5.32	1.98	**0.98**
	4	4.83	1.62	**0.80**
Two-point calibration	1	2.43	**0.69**	-
	2	0.84	**0.28**	0.36
	3	0.38	**0.11**	0.23
	4	0.21	**0.06**	0.22

rates were a bit smaller than those of two-point calibration, but we can still observe an increase from 47.4% to 72.6%. One interesting point is that the application of the prediction method described in a chapter 2.3.3(d) didn't improve the results when two-point calibration was used or when the estimation results without prediction were already good. On the contrary, when the error bounds are quite large (e.g., single-point calibration), we can reduce the errors quite effectively by applying the prediction method as shown in Table II.

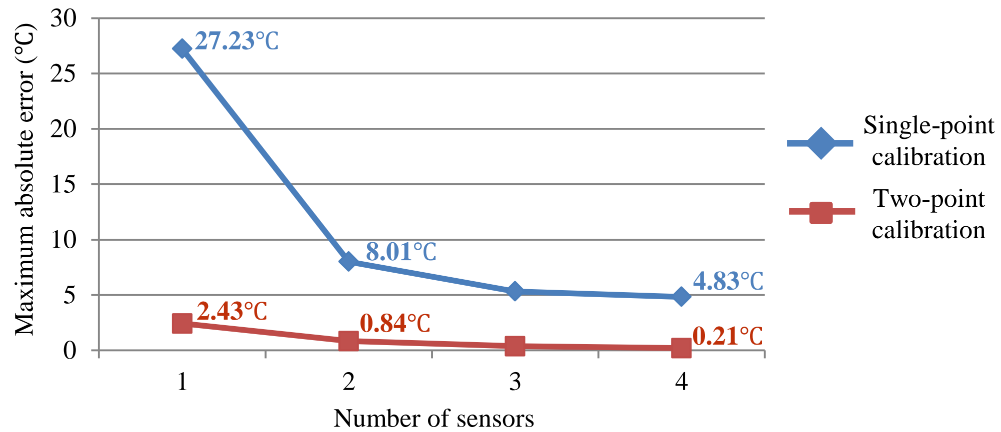


Figure 2.7. Maximum absolute errors as a function of the number of sensors

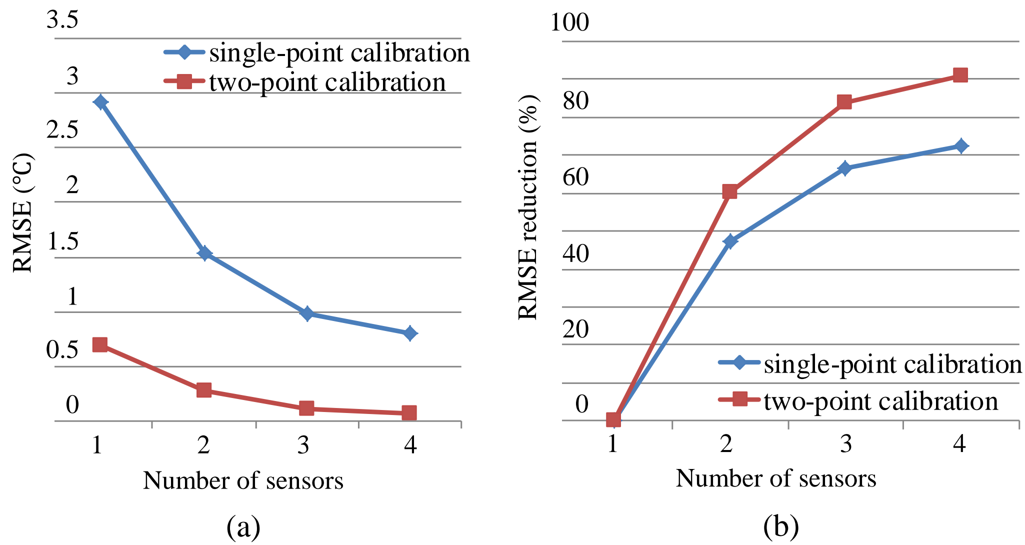


Figure 2.8. RMSE and its reduction rates as a function of the number of sensors

2.5. Summary

In this chapter, we proposed a method of obtaining accurate temperature information at one sensor location using multiple thermal sensors. By applying a sensor switching method and a prediction method, we were able to reduce the RMSE of 130nm RO-based thermal

sensors by up to 91.1% compared with the two-point calibrated single sensor case. We also showed that multiple sensors could be simulated by switching the calibration points of a single physical sensor in case the temperature at the sensor location changes slowly, which is true in most cases.

CHAPTER 3. Thermal Sensor Allocation for SoCs Based on Temperature Gradients

3.1. Motivation

DTM solutions need thermal distribution of a chip in order to manage and control chip temperatures properly at runtime. As discussed in previous chapters, there have been researches about using the performance counter or similar to estimate the thermal distribution of a chip instead of using dedicated thermal sensors [35]. The relationship between power and temperature is quite complicated and involves a lot of factors such as on-chip thermal diffusion and physical characteristics of silicon. As a result, it is easily observed that the region with the maximum power density is not the hottest part of a chip. That makes it more necessary to measure temperatures in different regions of a chip directly using dedicated on-chip thermal sensors than to estimate temperatures indirectly from power information.

In an ideal case, spreading a lot of accurate thermal sensors on a die will give satisfactory temperature information. However, it is not practical to use as many thermal sensors as possible because each sensor occupies valuable die area and also consumes power. The challenging question is how to use a small number of thermal sensors to acquire the accurate information about the thermal profiles of a chip at runtime. In this chapter, we address next two things: how many sensors are required and where should be their placement in order to

estimate or reconstruct thermal profiles accurately? How can we reconstruct full-chip thermal profiles in a reliable way?

3.2. Related work

Frequency domain analysis based on DCT (Discrete Cosine Transform) was proposed in [48] as briefly discussed in chapter 1, and it was based on an idea that a non-sparse signal in spatial domain could be transformed into a sparse signal in frequency domain with mostly zero coefficients; that is, as long as we can find major low frequency components correctly through proper sensor allocation, we can estimate any thermal profiles with high accuracy. This method works quite well when there are no major high spatial frequency components in a thermal profile. In other words, this method is good for relatively simple thermal profiles. However, with a lot more processing elements in a chip, thermal profiles of recent chips tend to have higher spatial frequency components than before, and as a result, a lot more thermal sensors need to be used to detect all major high frequency components and to maintain good full-chip reconstruction results.

As a novel approach, we propose using image processing and computer vision techniques in selecting the proper number of thermal sensors and their locations in order to reconstruct full-chip thermal profiles more accurately and efficiently in this chapter.

3.3. Thermal sensor allocation

The basic idea of the proposal is as follows; thermal profile of a chip has its own signature even though it keeps changing at runtime depending on the use of functional blocks in a chip over time. By analyzing the signature of the thermal profiles, we can collect basic information on its geometrical structure or framework, and this information can play an important role in estimating or reconstructing thermal profiles of a chip more effectively using just a small number of thermal sensors.

The rest of this proposal is organized as follows; in the rest of this section, details of our sensor allocation method and its results are given; in 3.4, profile reconstruction methods based on the sensor allocation results are explained; in 3.5, simulation results on profile reconstruction are reported.

3.3.1. Reference thermal profile generation

As we can see in Figure 3.1, which summarizes our thermal sensor allocation scheme, the first step is to generate a reference thermal profile of a chip. Various kinds of methods can be used for this, and one possibility is to use thermal simulation tools and the power estimates of each and every functional block on a die. In this proposal, we generate a reference thermal profile of a chip by averaging thousands of thermal profiles that we can collect by running various kinds of benchmark suites on a chip. We used thermal profiles from a thermal imaging device [50] instead of the results from thermal simulation tools in order to verify the effectiveness of our methods on real thermal data, and one exemplary reference thermal profile is given in Figure 3.2. A reference thermal profile generated from thermal simulation

tools will be also used at the end in order to see if our proposal also works quite well on those profiles. This process is to be performed just once for a chip, and it doesn't need to be performed at runtime.

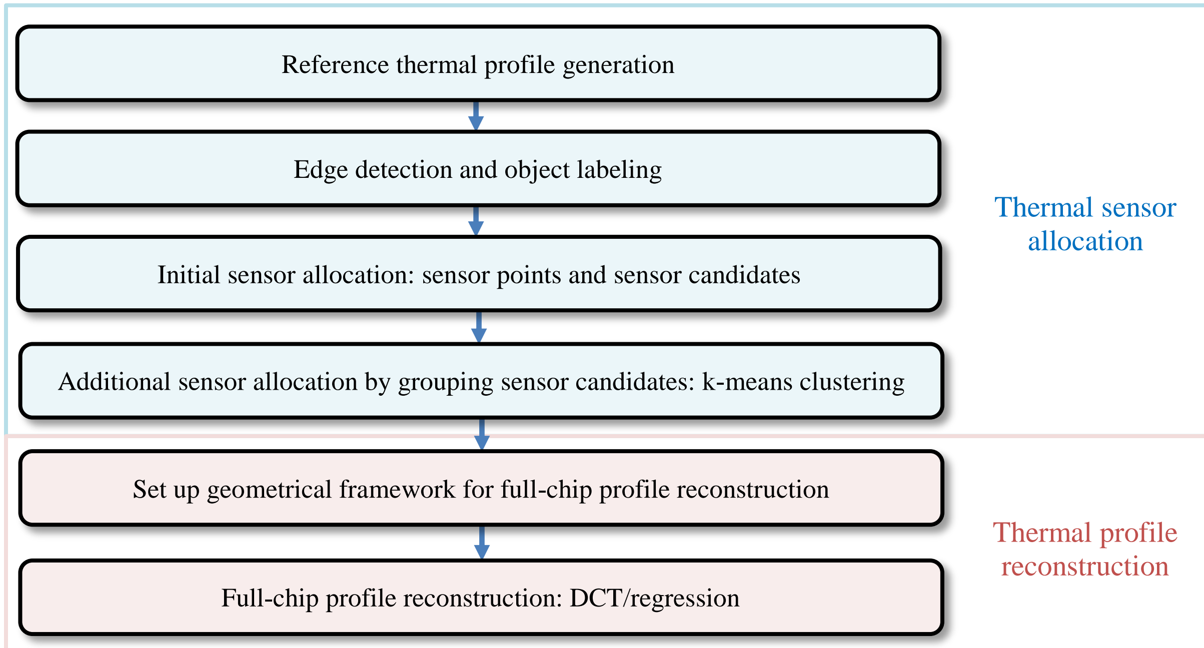


Figure 3.1. Steps for thermal sensor allocation and full-chip profile reconstruction

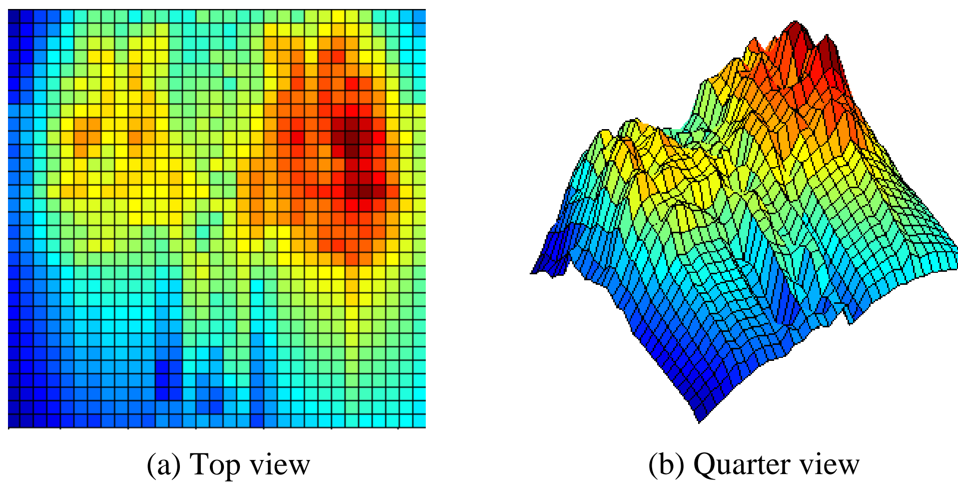


Figure 3.2. An example of reference thermal profiles of a chip

3.3.2. Edge detection and object labeling

The second step is to perform the analysis of the reference thermal profile and to extract meaningful objects from the profile using edge detection techniques such as Canny edge detector [73]. Here, edges are the sets of nodes on the grid with relatively high thermal gradients, and they usually correspond to the boundaries of local maxima or local minima, which compose the crucial elements in the geometrical framework of the profile. Mathematical morphology [74] is applied to the detection result to remove minor objects based on the size and the shape of each object, and the resulting objects are labeled and numbered as separate objects. Also some modifications such as trimming of a long and narrow object were applied to some objects based on the size and shape of each object so that we can make analysis of each object more properly in a later step. The result is given in Figure 3.3, and it presents eight separate objects in different colors.

Before we move on to the third step, we will define a few terms for clarification and will use them throughout this paper. Based on the aspect ratio of the length of minor axis to the

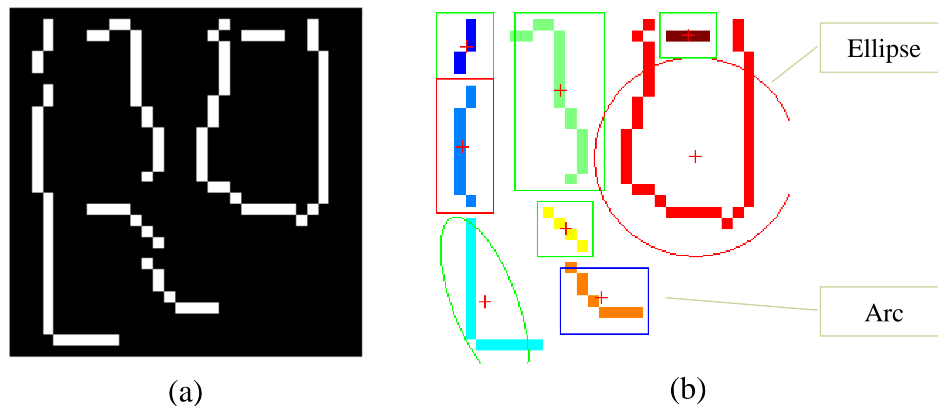


Figure 3.3. (a) Edge detection, (b) object labeling and analysis

length of major axis, each object is classified as either an ellipse or an arc; if the ratio of an object is greater than a pre-defined threshold value, then the object is classified as an ellipse; otherwise, it is classified as an arc. In our experiments, we use the value of 0.33 as the threshold, and the value was selected empirically through the simulations based on multiple thermal profiles of several different chips. When we analyze each object, we assign sensor points and sensor candidates to each object; a sensor point represents a node where a thermal sensor is placed; a sensor candidate, on the contrary, represents a node where a thermal sensor might be placed. All sensor candidates are compared with each other, and some of them are promoted to sensor points, and thermal sensors are assigned to them. The remaining sensor candidates will be still used as key elements composing the geometrical framework of the profile, and the temperature of each sensor candidate is determined based on the temperature readings of thermal sensors at sensor points and the geometrical framework.

3.3.3. Initial sensor allocation

a) Sensor points

In the third step, we figure out the best sensor locations for each object through heuristics, and we take slightly different approaches depending on the type of an object as explained in Figure 3.4.

In case of an ellipse, the hottest node and the coldest node inside the ellipse are searched for, and then the one with a shorter distance from its centroid than the other is selected as its sensor point.

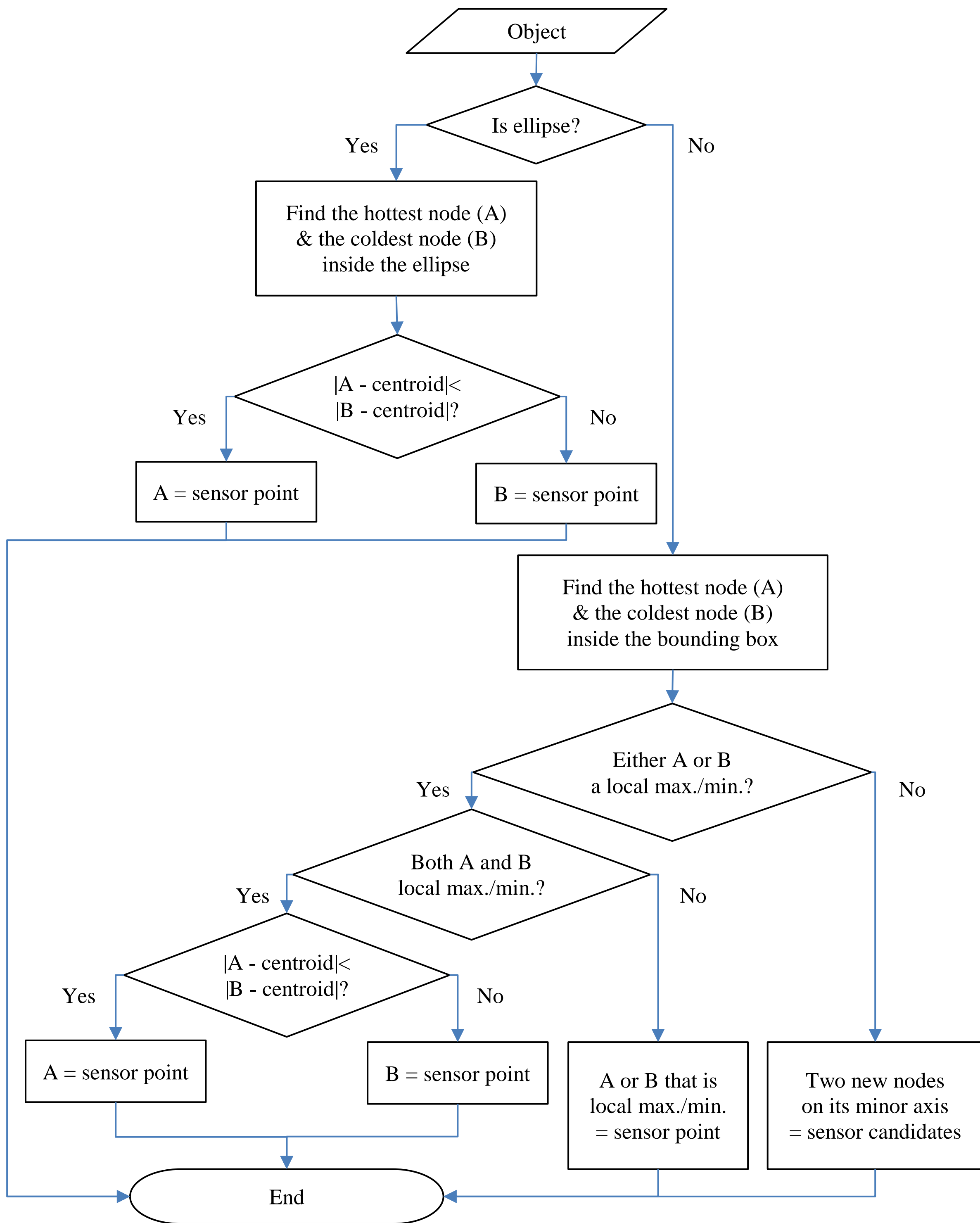


Figure 3.4. Sensor point allocation

In case of an arc, a corresponding bounding box is drawn first, and then the hottest node and the coldest node are searched for inside the box; if both nodes are either a local minimum or a local maximum, then we choose one whose Euclidean distance from the center of the bounding box is shorter than the other and assign the node as its sensor point; if just one of the two nodes is either a local maximum or a local minimum, then we assign the node as the sensor point of the arc; if neither of the two nodes is a local maximum nor a local minimum, then we discard both nodes and assign two new nodes at both ends of its minor axis as two sensor candidates of the arc instead, not selecting a sensor point of the arc.

b) Sensor candidates

In case only one node is assigned as the sensor point to an object in above sensor point assignment step, additional nodes are assigned as sensor candidates based on the geometrical properties of the profile, as explained in Figure 3.5.

In case of an ellipse, we draw horizontal and vertical lines crossing its centroid and check the gradient changes along those four line segments extending from the centroid. For each line segment, two nodes whose gradient values are the closest to 10% of the maximum gradient value along the segment, one inside and the other outside the ellipse, are assigned as its sensor candidates. Therefore, up to eight nodes can be assigned as sensor candidates to each ellipse. In order to handle a little bit differently, we define sensor candidates inside the ellipse as pseudo sensor candidates.

In case of an arc, one node, which is located on the other side of the object and at the same distance from the center of the bounding box, will be assigned as a sensor candidate.

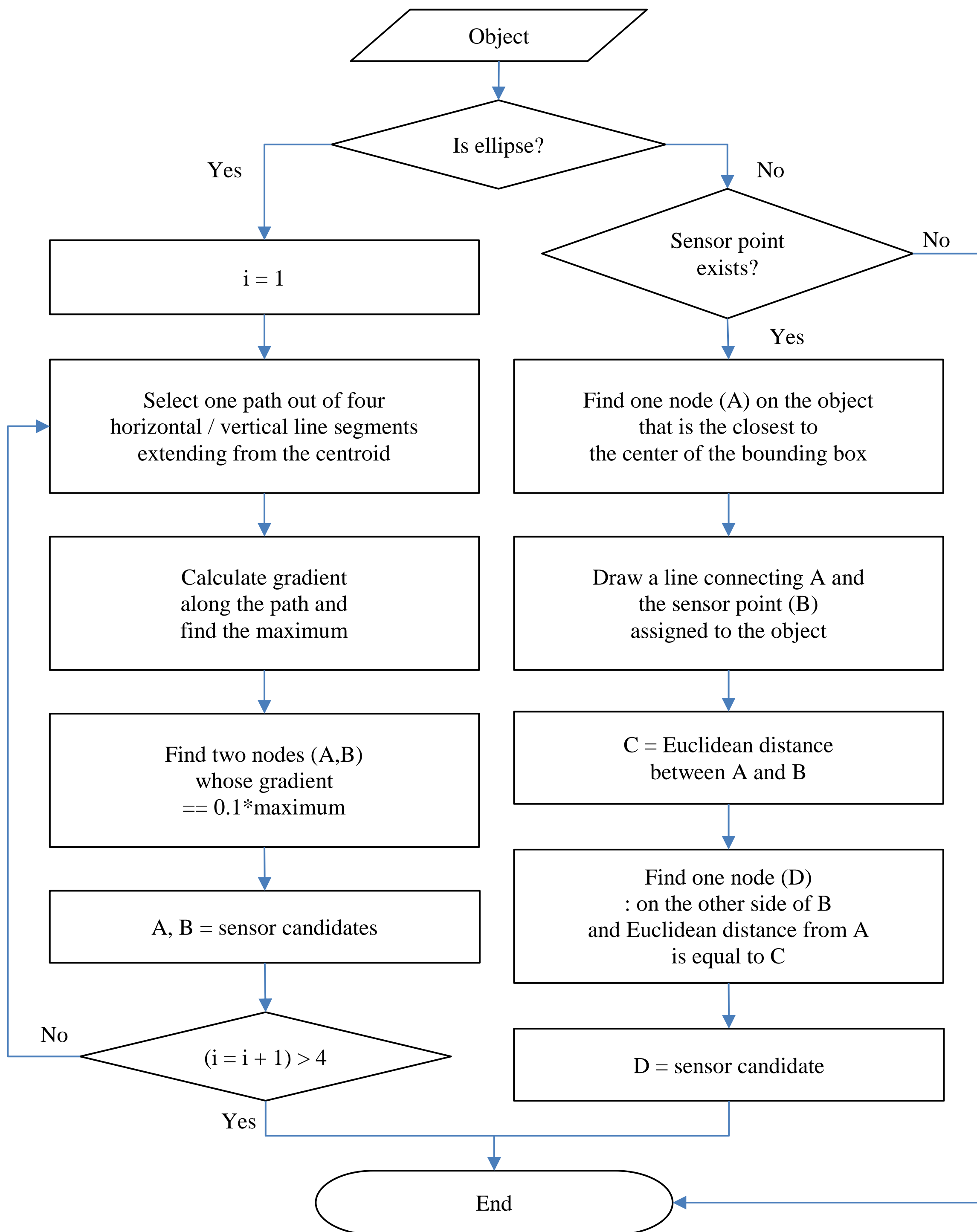


Figure 3.5. Sensor candidate allocation

One example with three sensor points and multiple sensor candidates, which is based on a reference thermal profile given in Figure 3.2, is given in Figure 3.6(a).

3.3.4. Additional sensor allocation

In order to select a few nodes among the multiple sensor candidates for additional thermal sensor allocation, we use k-means clustering method [43] to group sensor candidates into k clusters. Weighted sum of Euclidean distance and the temperature value is used as a metric with the ratio of two to five, which was chosen empirically. Normally, we choose the value of k to be equal to the number of sensor points, but we can flexibly change the number of thermal sensors to be allocated on a die by changing the value of k. One example of k = 6 is given in Figure 3.6(b), and all sensor candidates in each cluster are marked in the same color and will be assigned the same temperature value. Finally, we need to select a representing sensor candidate node for each cluster to allocate a thermal sensor, and we pick one whose

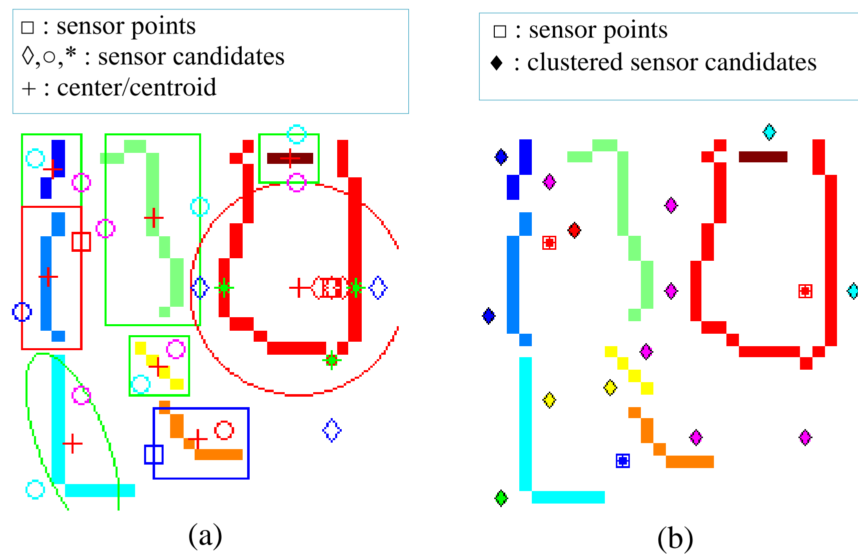


Figure 3.6. (a) Sensor points and sensor candidates (including pseudo candidates), (b) k-means clustering (k = 6) on sensor candidates

temperature value is the closest to the mean of the temperature values of all sensor candidates in the cluster. In k-means clustering, pseudo sensor candidates are not included, and the temperature values of each pseudo sensor candidate node are set equal to the temperature values of a thermal sensor assigned to its corresponding sensor point.

3.3.5. Geometrical framework

Even though we set up a basic geometrical framework using multiple sensor points and sensor candidates, it would be still helpful to add some additional nodes to the framework based on the geometrical information of all objects for better profile reconstruction.

In case of an ellipse, we choose up to four additional nodes on the object in four directions, whose temperature values are set to be the mean of the two corresponding sensor candidates inside and outside the ellipse.

In case of an arc, we basically use all nodes on the object running parallel to the major axis, and the mean of the temperature values of two sensor points or candidates assigned to the object is used as their temperature values.

One exemplary geometrical framework of the reference thermal profile that we used in previous steps is given in Figure 3.7, for the case of $k = 6$; the framework is composed of nine thermal sensor nodes, three from sensor points and six from sensor candidates, and additional 23 nodes that were selected based on the geometrical information of all objects on the profile. The temperature values of those 32 nodes are assigned based on the temperature readings of nine thermal sensors and the geometrical information of the profile.

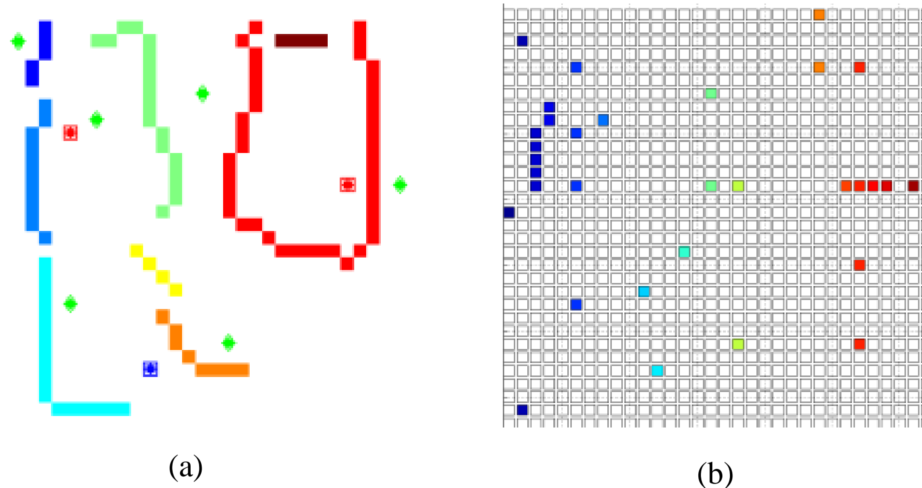


Figure 3.7. (a) 9 thermal sensors (3 from sensor points and 6 from sensor candidates) (b) geometrical framework composed of 32 nodes

3.4. Thermal profile reconstruction

In this proposal, we use both a regression method [75] and a transform-based method for full-chip profile reconstruction over an entire die area. In case of a transform-based method, DCT [49] is used for the transformation, and the details are explained in Figure 3.8 and Figure 3.9 [49]; we transform 2D signals in spatial domain, i.e., thermal profiles from sensor readings, to their counterparts in frequency domain using DCT, and we perform low pass filtering in frequency domain. Low pass filtered results are then transformed back to the ones in spatial domain using similar matrix calculations. A matrix $S_{sensors}$ with the dimension m by n is generated from a full-size matrix by removing rows and columns whose elements are all zeroes. Because this m by n matrix is a sparse matrix, we use a regression technique to replace its zero elements with non-zero values before we calculate $F_{LowFreq1}$ for better reconstruction results. For low pass filtering in frequency domain, we select ‘N-1’ low frequency coefficients

$S_{estimated} = T' \cdot F_{LowFreq2} \cdot T$

, where $F_{LowFreq2}$ is a full-size matrix from $F_{LowFreq1}$, with zero padding

and $F_{LowFreq1} = (A'A)^{-1} \cdot A' \cdot S_{sensors} \cdot B' \cdot (BB')^{-1}$

- $S_{estimated}$ (full-size): estimated thermal profiles (spatial domain)
- T (full-size): DCT matrix
- $F_{LowFreq1}$ $(k \times t)$: thermal profiles with low frequency coefficients only (freq. domain)
- $S_{sensors}$ $(m \times n)$: thermal profiles from sensor readings (spatial domain)
- A $(m \times k)$: sampled matrix from T'
- B $(t \times n)$: sampled matrix from T

Figure 3.8. Transform-based reconstruction

in a zigzag pattern [49], where *N* is the total number of nodes in the geometrical framework with non-zero temperature values.

One thing to note is, when we consider sensor locations as sampling points, sampling of thermal profiles is clearly non-uniform. Therefore, it is necessary to figure out carefully how to transform the non-uniformly sampled profiles between spatial domain and frequency domain, especially when the number of sampling points is quite limited. However, with the additional nodes in the geometrical framework other than the thermal sensor nodes, we can have good results even when we use a simpler transformation method than the ones used in previous methods.

grid_sz = 64 or 32, depending on the dimension of thermal profiles;

T = DCT matrix, grid_sz by grid_sz;

for (i = 1; i <= grid_sz; i++)

for (j = 1; j <= grid_sz; j++)

if i == 1

$$T(i^{th}\ row, j^{th}\ column) = \sqrt{\frac{1}{\text{grid_sz}}};$$

else

$$T(i^{th}\ row, j^{th}\ column) = \sqrt{\frac{2}{\text{grid_sz}}} * cos\left(\frac{(2j-1)*(i-1)}{2*\text{grid_sz}}\pi\right);$$

Figure 3.9. DCT matrix generation

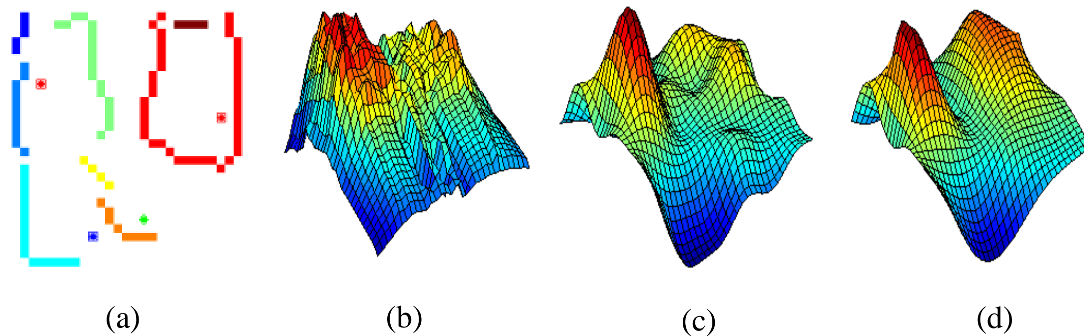
3.5. Experimental results

For the evaluation, we used 7,273 thermal profiles of AMD Athlon2 X2 240 processor, which were captured by a thermal imaging device, running SPEC2006 benchmark suites [50]. The maximum temperature on the processor was 47.65℃, and the difference between the maximum and the minimum was 9.29℃ on the average, ranging from 1.62℃ up to 16℃. Using 4 to 16 sensors, with the value of *k* ranging from 1 to 13, we reconstructed all of the 7,273 profiles purely based on sensor readings and the geometrical framework that we built, and then we calculated two metrics for each profile: RMSE over all nodes on the grid and the absolute error at the hottest spot, which was normalized by the temperature difference between the hottest node and the coldest node.

Exemplary reconstruction results of the case k = 1, or the case of using four thermal sensors, is given in Figure 3.10, and we can observe the results of our methods are quite similar to the true profile to be estimated even though just four sensors were used.

Performance comparison of two reconstruction methods, DCT and regression, along with the method in [48] is given in Figure 3.11 and Figure 3.12. Averaged RMSE over 7,273 profiles is given in Figure 3.11, and the averaged absolute error at the hottest spot over all the profiles is given in Figure 3.12.

The method in [48] solely depends on the sensor readings in its reconstruction; thus, the RMSE over the grid and the absolute error at the hottest spot were both worse than our methods, especially when the number of sensors was quite limited. In addition, we observed that the absolute error at the hottest spot increased even with a larger number of sensors in case sensor locations were not close to the hottest spot. On the contrary, our approach achieved improvements of up to 42% in RMSE and also up to 93% in hottest spot estimation when compared with the method in [48], and it is mainly due to the fact that we used both sensor readings and the geometrical framework in an effective and cooperative way for the



(a) (b) (c) (d)

Figure 3.10. Sensor allocation and profile reconstruction results when the number of sensors is set to four (or k = 1) (a) sensor allocation result, (b) profile to be reconstructed, (c) reconstruction: DCT, (d) reconstruction: regression

reconstruction of thermal profiles.

Comparing two reconstruction methods that we used, DCT and regression, we can see the DCT-based reconstruction method achieved a little bit better results when it comes to RMSE, and slightly worse results in hotspot temperature estimation. Considering the higher computational complexity of a regression method, a DCT-based reconstruction method would be a reasonable choice maintaining a good balance between RMSE and hotspot temperature estimation errors.

In order to verify if the proposed method works equally well on other chips, especially

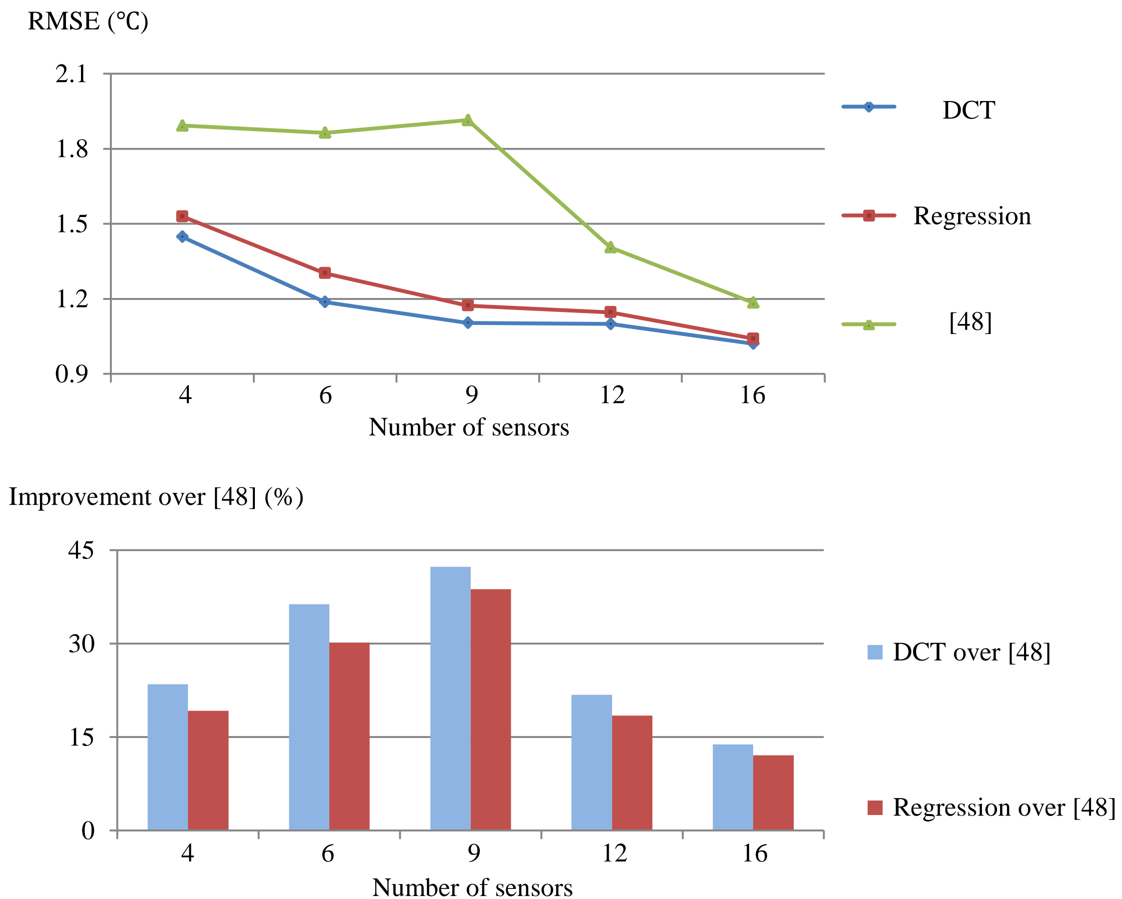
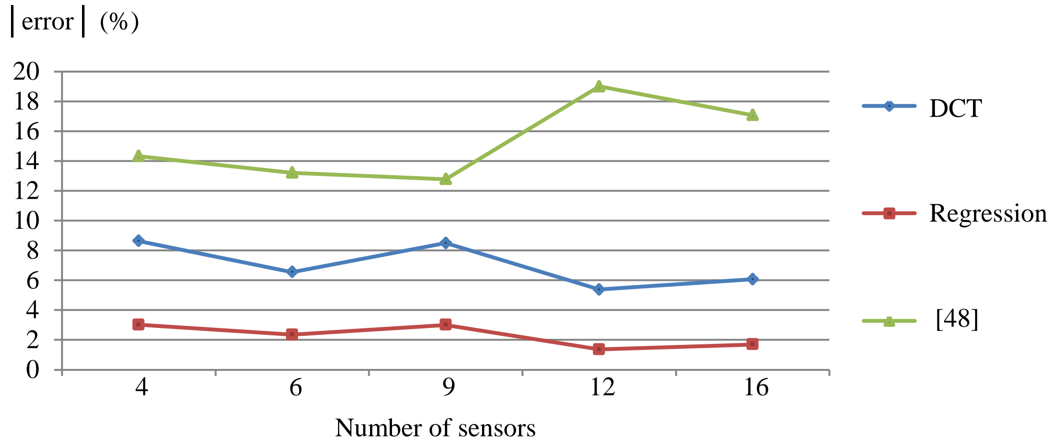


Figure 3.11. RMSE as a function of the number of sensors



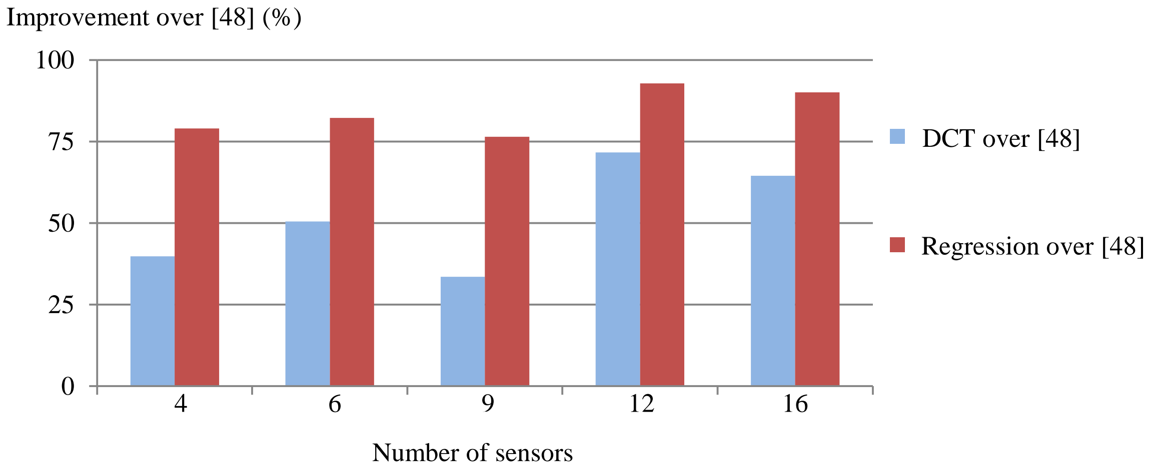


Figure 3.12. Absolute error at the hottest spot

when the representative thermal profiles of the chips are generated from thermal simulation tools, we applied the exactly same method without any modification to a new industrial-size SoC that was composed of six core clusters and a large number of functional blocks. A reference thermal profile of the SoC was generated from a thermal simulation tool HotSpot [76], based on the worst case power consumption of each and every functional block on a die, and 63 additional thermal profiles were generated from HotSpot for a test purpose, switching on and off six core clusters. Each thermal profile was presented on a grid of 64 by 64, with

4096 nodes in total on the grid. The maximum temperature over the 64 thermal profiles was 105.65℃, and the minimum was 90.47℃. The averaged maximum temperature over the 64 profiles was 102.76℃, and the averaged minimum was 93.02℃, with the mean temperature difference of 9.74℃. For comparison purpose, results of each step were prepared and given in four figures; a reference thermal profile of this new SoC is given in Figure 3.13; edge detection and object classification results are given in Figure 3.14; assignment of sensor points and sensor candidates is given in Figure 3.15; final sensor allocation results and the geometrical framework are given in Figure 3.16.

Nine thermal sensors, six from sensor points and three from sensor candidates (k = 3), were used for the reconstruction of thermal profiles as given in Figure 3.16. The averaged RMSE of a DCT-based reconstruction method was 0.93℃, and the averaged RMSE of a regression-based reconstruction method was 0.75℃, achieving 43% and 54% improvement over [48] whose averaged RMSE was 1.63℃. When we considered 1024 hot nodes out of 4096 nodes, whose temperatures were within top 25% when sorted in the order of temperature,

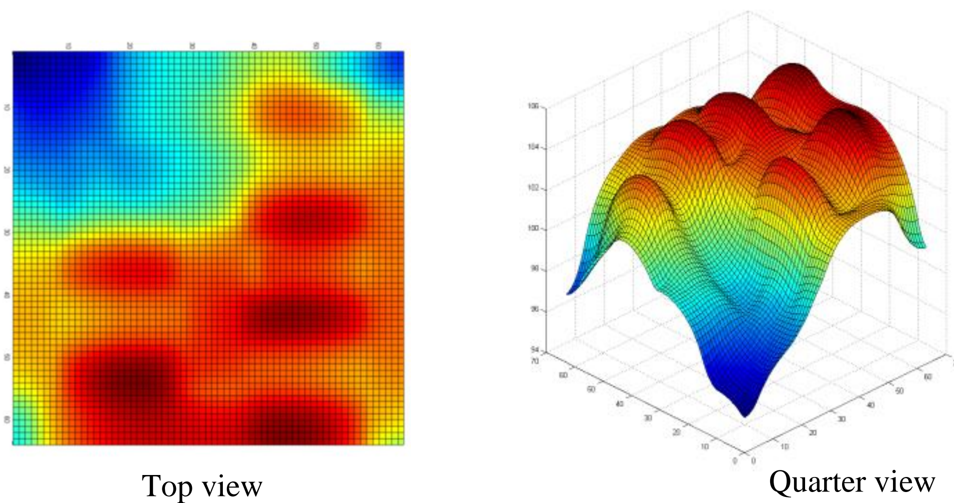


Figure 3.13. Reference thermal profile of a new chip

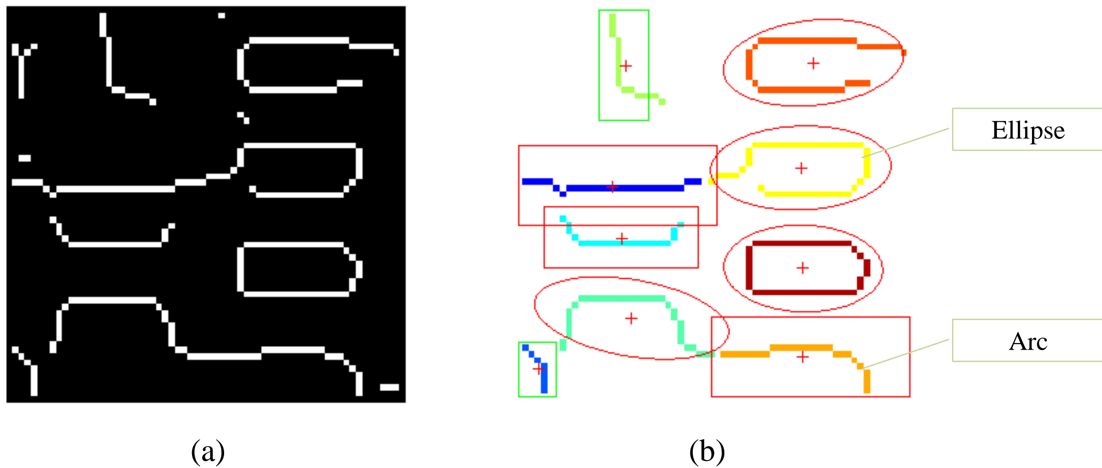


Figure 3.14. (a) Edge detection, (b) object labeling and analysis of a new chip

the averaged RMSEs of both a DCT-based reconstruction method and a regression-based method dropped to 0.69°C. This is mainly due to the fact that no thermal sensors were assigned to the nodes near the cold spots on the profile, and as a result, the errors tended to increase in the estimation of low temperature nodes. Averaged RMSEs of three methods over 64 thermal profiles are summarized in Figure 3.17, and it is clear that both of our methods gave better results than the method in [48], especially when it comes to the temperature estimation of hotspots.

Just like RMSE cases, the method in [48] gave much worse results when it comes to the absolute errors at the hottest nodes on thermal profiles. A regression-based reconstruction method resulted in an averaged absolute error of 0.01°C, and a DCT-based method resulted in an average value of 0.11°C, while the method in [48] gave an average value of 1.74°C.

The method in [48] might have given good results with the use of a large number of thermal sensors. However, it was simply not enough to use just nine thermal sensors to capture all spatial frequency components of the profiles, which are necessary for the accurate

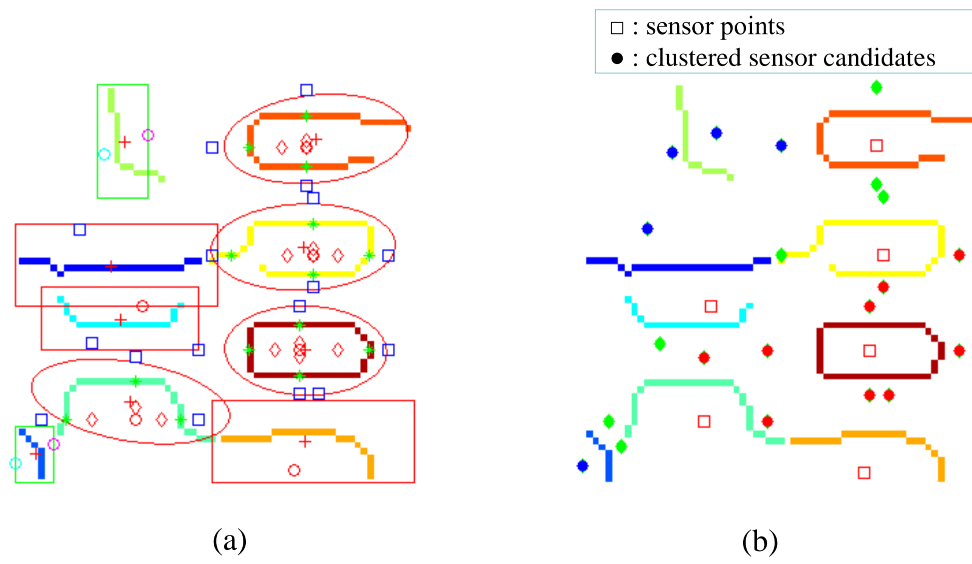


Figure 3.15. (a) Sensor points and sensor candidates of a new chip, (b) *k*-means clustering on sensor candidates ($k = 3$) of a new chip

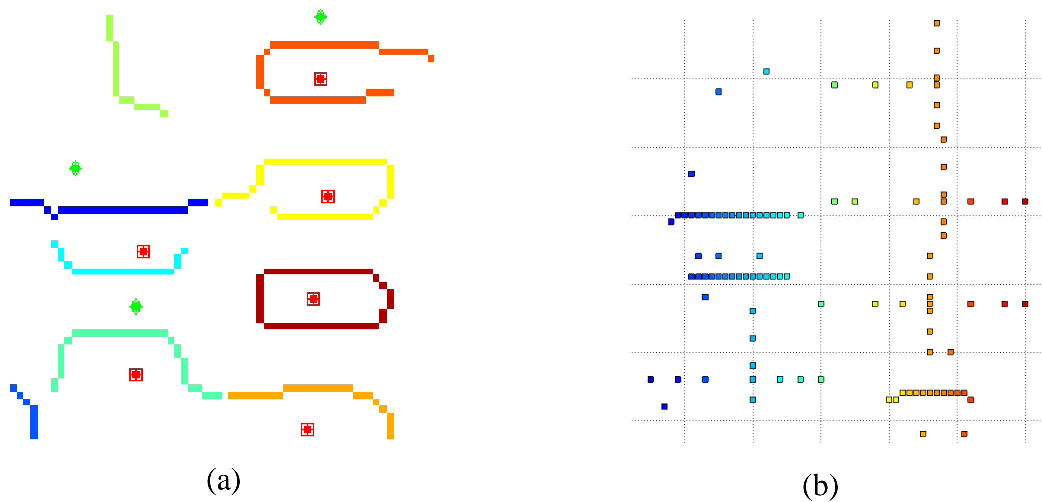


Figure 3.16. (a) 9 thermal sensor nodes of a new chip, (b) geometrical framework composed of 105 nodes to which temperature values are assigned

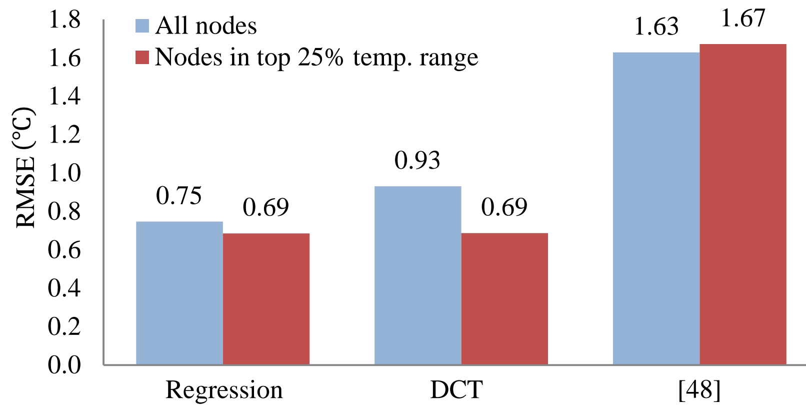


Figure 3.17. RMSE comparison among three methods: considering all 4096 nodes, and considering 1024 hot nodes only

profile reconstruction. On the contrary, our approach used the temperature values at 105 nodes out of 4096 nodes, which were all derived from both the temperature readings of nine thermal sensors and the geometrical framework of the profile. As a result, both a regression-based method and a DCT-based method worked quite well just using nine thermal sensors.

Regarding the comparison between the results of AMD X2 240 processor and a new SoC with six core clusters, the first observation that we can make is the increase in the number of cores or core clusters from two to six resulted in the increased number of thermal sensors; the number of sensor points increased from three to six. Another observation to be made is that the averaged RMSE of each method decreased when compared with its counterpart of a dual-core case, and this is mainly because the profiles generated from HotSpot were quite smoother than the noisy profiles captured by a thermal imaging device, and also because the variation among the 64 profiles was smaller than the variation among the 7,273 profiles. We can also observe that the improvements over [48] in percentage are higher than the corresponding figures of a dual-core case with a similar choice of *k*. This is due to the fact that the method in

[48], which purely depends on the temperature values at the sensor nodes for the reconstruction of thermal profiles, requires a lot more thermal sensors than the two proposed method do when thermal profiles become quite complicated with a lot of high spatial frequency components.

3.6. Summary

In this chapter, we proposed new methods of thermal sensor allocation and full-chip thermal profile reconstruction through the analysis of thermal profiles. We considered a reference thermal profile of a chip as an image and applied various image processing and computer vision techniques to extract its geometrical framework. Then we used the framework to reconstruct thermal profiles from the temperature readings of a small number of thermal sensors efficiently. We believe that our methods are more efficient and may be more suitable than others, especially when thermal profiles are complicated due to multiple cores on a die for emerging SoCs.

CHAPTER 4. Vision-inspired Global Routing for Enhanced Performance and Reliability

4.1. Motivation

While Moore's law has enabled the advent of products with increasing functionality and complexity, shrinking geometries and increasing power densities have led to higher operating temperatures of chips. In addition, due to different switching activity rates and different types of circuits, different parts of a chip have different power densities, and as a result, they will have different temperatures. For example, the power density of the integer processing unit will be much higher than that of a cache memory. Furthermore, the low thermal conductivity of silicon will make the lateral heat propagation rate slow, which will cause localized heating. These non-uniform power densities and the low thermal conductivity of silicon will result in a thermal distribution that varies from one part of a chip to another and thermal hotspots that are caused by localized areas of high power densities.

As we discussed in previous chapters, the temperature difference within an SoC can be as high as 50 °C across the die [4] [5], and this non-uniform thermal distribution along with high operating temperatures can cause a large number of issues; reduced life expectancy of interconnects due to electromigration [6]; system degradation caused by lowered clock frequencies to prevent any delay-induced failure; increased leakage power, etc. In this chapter,

we focus on the issues of the life expectancy and reliability of interconnects and the increased delay due to non-uniform thermal distribution.

The interconnect delay model in the deep submicron era, which depends on the thermal distribution of substrates, was derived as [18]:

$$D = D_0 + (c_0 L + C_L)\rho_0\,\beta \int_0^L T(x)dx - c_0\rho_0\,\beta \int_0^L xT(x)dx \qquad (4.1)$$

In the equation, D_0 is the Elmore delay of the interconnect corresponding to the unit length resistance at 0°C, and the detailed discussion was given in chapter 1.

In this chapter, we propose a method of using the thermal distribution of substrates and the temperature-dependent interconnect delay given by (4.1) to select optimal paths out of multiple candidates and also to minimize the delays and the number of wires near hotspots for better reliability.

4.2. Related work

Several approaches have been proposed for temperature-aware global routing recently. TAGORE [58] uses an iterative L-shaped pattern routing in order to reduce the number of nets passing through hotspots and then uses a maze router [77] to route the remaining nets. Basically, TAGORE considers only two simple L-shaped paths for a given two-terminal net, and maximum temperatures along the paths are used as a metric; maximum temperatures of those two paths are compared with each other, and a path with a lower maximum temperature

is chosen as a path for the given net. While TAGORE decreases the worst-case failure rate by selecting a path with a lower maximum temperature, it doesn't consider the delay of each path. As a result, it is possible that the selected path has a larger delay than the other, especially when the thermal distribution along the selected path is constantly higher than the other. In addition, routers using larger solution space will give better results than TAGORE since it considers only two possible paths for each net. In [78], two thermal-driven techniques were proposed as ways of reducing the probability of interconnect failure: thermal-driven MST (Minimum Spanning Tree) construction and thermal-driven maze routing. It achieves better results when compared with TAGORE mainly because it explores much larger solution space using maze routing, but the better results come with higher computational complexity.

4.3. Vision-inspired global routing

As a novel approach, we propose using Image processing and computer vision techniques for global routing, and the detailed description is given in this section.

4.3.1. Overall flow

The overall flow of our proposed method is given in Figure 4.1. We generate a thermal profile of a given chip using a thermal simulation tool HotSpot [76] and then we detect major peak and valley areas, or local maxima and minima, within the profile using image processing and computer vision techniques. We decompose each net from a given circuit into a set of two-terminal nets using Prim's algorithm [79], which is used to find a Minimum Spanning

Tree (MST) [79]. Each two-terminal net defines a window, and the collected information about the peak and valley areas on the profile is used to find a number of internal nodes within the window, which will be used for solution space expansion in a later step. Using the resulting set of internal nodes within the window, we draw horizontal and vertical lines passing through them and then assign additional nodes at the crossings between those lines and the boundaries of the window. We define each path connecting a pair of nodes as an edge and use the temperature-aware delay of each path as the weight of the edge. Using the two terminals and the resulting set of nodes as vertices and the horizontal and vertical paths as edges, we apply Dijkstra's algorithm [79] to find a path with a minimum delay out of all possible paths for each two-terminal net.

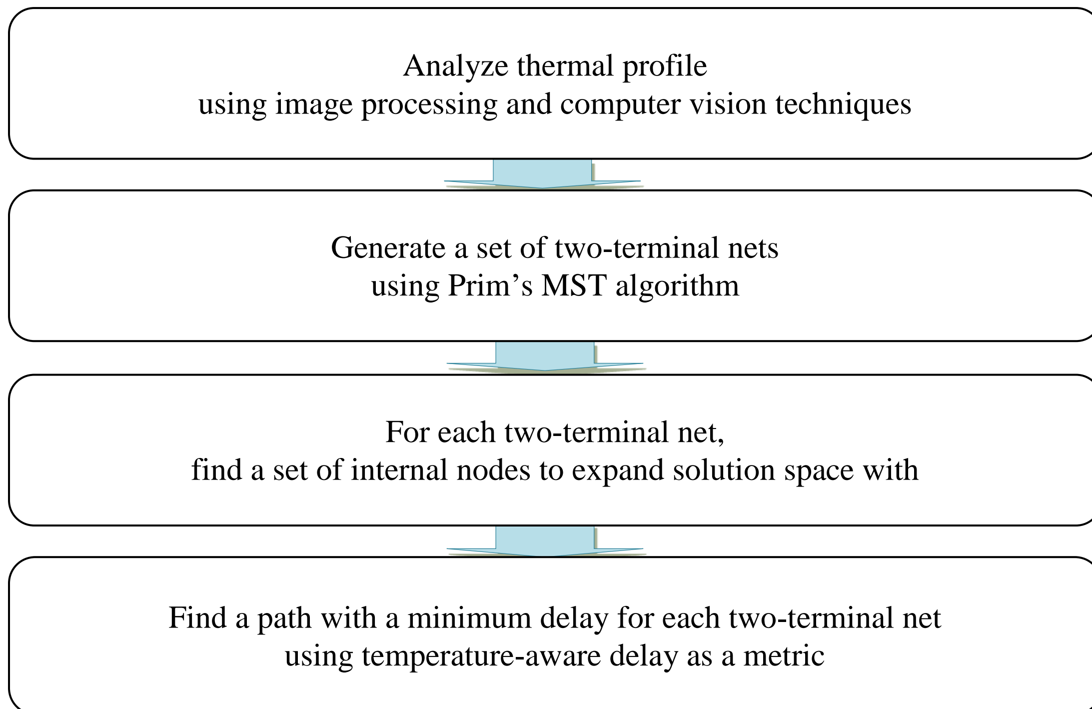


Figure 4.1. Overall flow of vision-inspired global router

4.3.2. Peak and valley detection

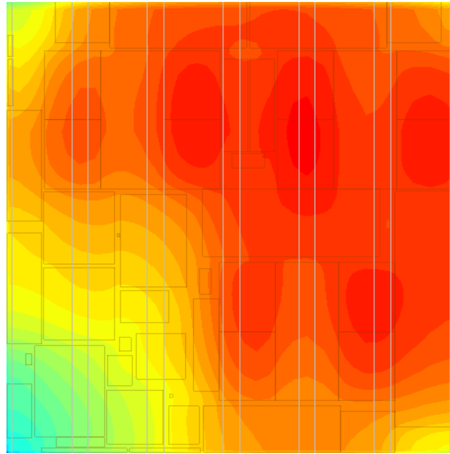


Figure 4.2. HotSpot [76]-generated thermal profile

Figure 4.2 shows a thermal profile from the thermal simulations on an industrial size SoC, and HotSpot [76] was used for the generation of the profile on a grid of 64 by 64 nodes. Temperature difference between the maximum and the minimum temperature within the profile is 45°C, and the maximum temperature is 105°C. We can locate multiple peak and valley areas within this thermal profile using image processing and computer vision techniques, and this information can be used to find paths with better reliability and minimized delay.

First, we can define an area with relatively lower temperatures compared with its surrounding areas as a valley, and this area can be used for the routing to avoid hotspots. Likewise, we define an area with relatively higher temperatures compared with its surroundings as a peak, and this area should be avoided especially in global routing to improve the reliability of the interconnects or the chip. We can locate peak and valley areas efficiently and easily using the gradient at each node, which is defined as follows:

$$\nabla T(x, y) = \frac{\partial T}{\partial x}\vec{x} + \frac{\partial T}{\partial y}\vec{y} \qquad (4.1)$$

In the equation, $T(x, y)$ is the temperature of a node at (x, y), and $\vec{x}$ and $\vec{y}$ are the unit vectors pointing in the positive x direction and the positive y direction, respectively. The gradient at each node of the thermal profile, which is a vector, is shown in Figure 4.3 as a small arrow with its own magnitude and direction.

By comparing Figure 4.2 and Figure 4.3, we can easily see that the areas with small arrows, i.e., with $\frac{\partial T}{\partial x}$ and $\frac{\partial T}{\partial y}$ close to zero, represent peak areas or valley areas, depending on the directions of corresponding arrows. Therefore, the first step to identify peak areas and valley areas is to find nodes with small arrows on the gradient map, and the result is given in Figure 4.4, with all marked with small blue circles.

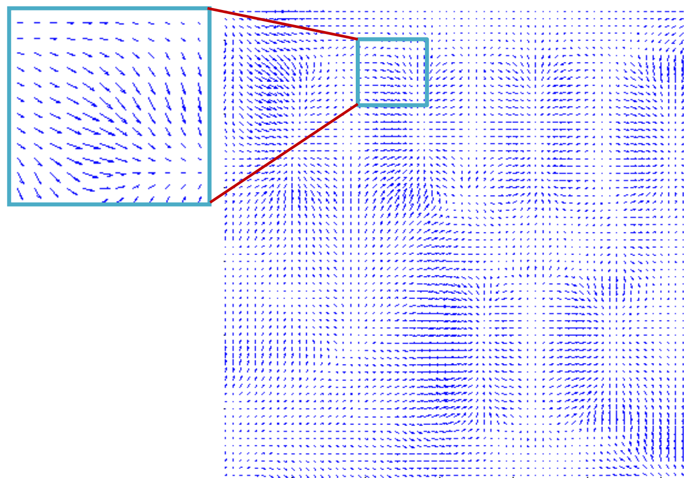


Figure 4.3. Gradient at each node on a grid of 64 by 64 nodes

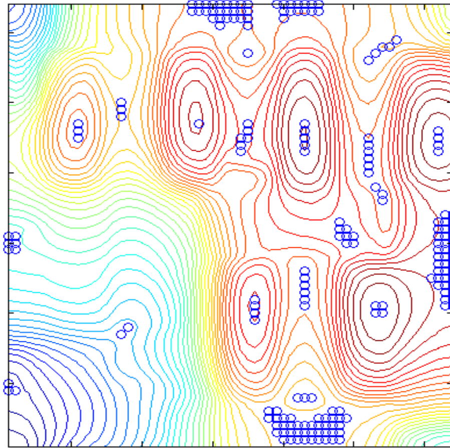


Figure 4.4. Peak and valley areas: identifying nodes with small gradient magnitude

Using k-means clustering [43] and a simple grouping method, we group nodes with small gradient magnitude into a number of separate areas, and then each area is classified into either a peak area or a valley area based on the gradient variation within each area. The result is given in Figure 4.5, and red rectangles represent peak areas, and cyan rectangles represent valley areas, respectively. In this figure, a larger rectangle than its actual size is used for clarification purpose.

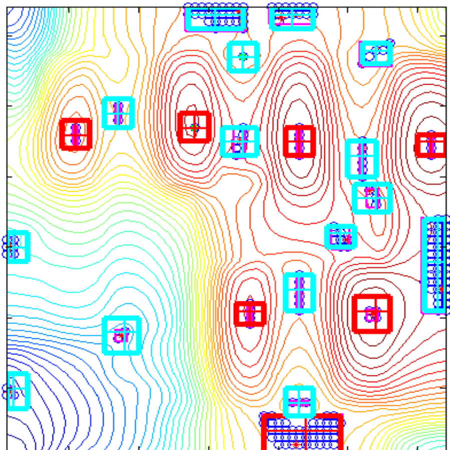


Figure 4.5. Peak and valley areas: classification

4.3.3. Solution space expansion

When a two-terminal net is given, we have a window defined by those two terminals. TAGORE [58] considers only two simple L-shaped paths on the boundaries of this window as routable paths of the given net. Therefore, its solution space is too small to find an optimal path for the net. A thermal-driven maze routing was used in [78] to explore much larger solution space, but the increasing number of nets and a large chip area of current chips make the use of maze routing quite burdensome computationally. Using the collected information on the locations of valley areas within a thermal profile, we can expand solution space efficiently to explore much more possible paths, but the solution space will be a lot more compact than the one used in [78].

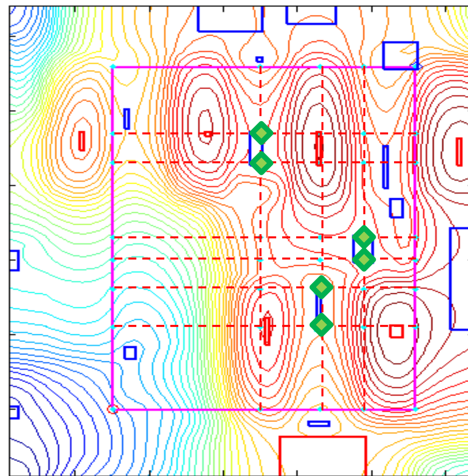


Figure 4.6. Solution space expansion by adding a limited number of internal nodes

In Figure 4.6, a rectangle with magenta boundaries is a window defined by a two-terminal net, and peak and valley areas inside the window are marked with red and blue rectangles, respectively. While a larger rectangle than its actual size was used for clarification purpose in Figure 4.5, peak and valley areas in figure 4.6 are given in their actual sizes. Basically, each

valley area within the window adds two internal nodes along the long axis, and we choose a proper number of internal nodes in order to expand the solution space suitably. In Figure 4.6, we limited the number of newly added internal nodes to six in order to prevent the solution space from becoming too large, and we chose six internal nodes from the three largest valley areas inside the window, two nodes from each valley area. Newly chosen six internal nodes are marked with green diamonds in the figure, and the new paths generated from these new internal nodes are given in dotted red. In this example, we have 38 new nodes in total inside the window and on the boundaries in addition to the two terminals that define the window.

If there are no valley areas inside a given window, we take a different approach to expand the solution space. First, we find the nodes within the window whose temperatures are lower than the minimum of the temperatures of the two terminals defining the window, and we group them into several separate areas using image processing techniques. We find the centroid of each area and choose a proper number of centroids as new internal nodes. We also limit the number of newly added internal nodes in order to prevent too large solution space.

4.3.4. Reliability and performance metric

We basically use the temperature-aware delay as a metric, and we choose a proper path connecting the two terminals of a given net using the metric. Because each two-terminal net will have a window in a different size at a new location and a new set of nodes, we need to calculate the weight of each edge before we use Dijkstra's algorithm [79] to find a path for the given net. In order to reduce the resulting computational complexity, we calculate

temperature-aware, node-to-node unit delays first so that we can approximate the weight of each edge by simply summing up the unit delays along the edge.

There might be some cases that a selected path, which was selected purely based on the delay of the path, passes through hotspots for a fraction of the path. In order to give more priority to better reliability than to delay minimization of the path, we compare the maximum temperature of a resulting path with the maximum temperatures of two L-shaped paths on the boundaries of the window. If the maximum temperature of the resulting path is higher than the minimum of the maximum temperatures of the two L-shaped paths on the boundaries of the window, then one of the two L-shaped paths with a lower maximum temperature is selected as a path for the given net instead.

The routing of a net given in Figure 4.6 is plotted as a dotted green line in Figure 4.7.

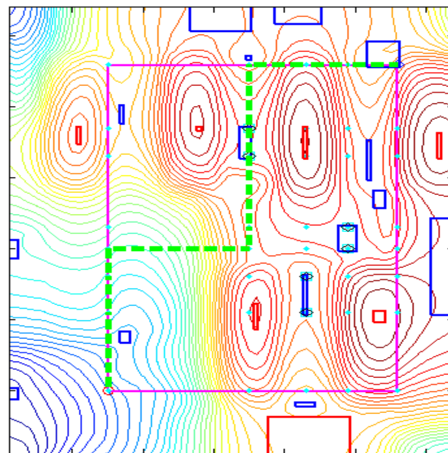


Figure 4.7. Routing of a net given in Figure 4.6

4.4. Experimental results

We implemented and evaluated our vision-inspired global router in C++ and MATLAB, and we used nine benchmarks [80] to show the effectiveness of our method. Thermal profiles for the benchmarks were generated by [76] on a grid of 64 by 64 nodes, and the number of nodes of each benchmark was modified properly so that it was equal to the number of nodes of a corresponding thermal profile. For comparison purpose, a conventional router and TAGORE [58] were implemented together. All three routers basically follow the same steps explained in section 4.3.1, but they choose a path for each two-terminal net in different ways; TAGORE chooses a path out of two L-shaped paths based on the maximum temperature of each path; a conventional router chooses a path randomly out of the two L-shaped paths. For a clear comparison, one-dimensional nets, of which two terminals lie on a same horizontal or vertical line, were not used for data collection, and only two-dimensional nets were used. The number of internal nodes within a window, which is defined by a two-terminal net, was limited to 15 in the experiment in order to limit the size of solution space.

We collected temperature information of the nodes on all resulting paths for each benchmark and then compared the accumulated number of nodes within hotspots whose temperatures lie between 104.1°C and 105°C, which is within top 2% of the entire temperature range.

The result is given in Table III. When it comes to the reduction in the number of nodes in hotspots, our router gave better results than a conventional router by up to 50% reduction, and 42% reduction on the average. When compared with TAGORE, our router reduced the number of nodes in hotspots by up to 24% and 13% on the average. We can also observe in

Figure 4.8 and Figure 4.9 that our router gave increasingly better results with the increase in the size of the circuits. This is because larger circuits will have larger windows and longer global interconnects, and our router can handle them a lot more efficiently than the others.

When it comes to the reduction in delay, our router reduced delay by up to 4.11% when compared with a conventional router. Even though the result in delay reduction was not so

Table III. Comparison in the number of nodes in hotspots and delay reduction

Circuits	# nets	# grids	**Accumulated # of nodes in hotspots** (reduction in # of nodes in comparison with a conventional router)			**Maximum delay reduction** (in comparison with a conventional router)
			Conventional router	**TAGORE** [58]	**Our router**	
IBM01	11507	64×64	490	320 (34.7%↓)	309 (36.9%↓)	2.17%
IBM03	21621	80×64	967	727 (24.8%↓)	658 (32.0%↓)	3.75%
IBM04	26163	96×64	946	542 (42.7%↓)	526 (44.4%↓)	3.29%
IBM05	27777	128×64	3345	2190 (34.5%↓)	1900 (43.2%↓)	3.98%
IBM06	33354	128×64	2021	1374 (32.0%↓)	1217 (39.8%↓)	4.11%
IBM07	44394	192×64	2178	1492 (31.5%↓)	1200 (44.9%↓)	3.22%
IBM08	47944	192×64	2910	2170 (25.4%↓)	1846 (36.6%↓)	3.38%
IBM09	50393	256×64	2437	1656 (32.0%↓)	1265 (48.1%↓)	3.66%
IBM10	64227	256×64	5802	3561 (38.6%↓)	2923 (49.6%↓)	3.78%

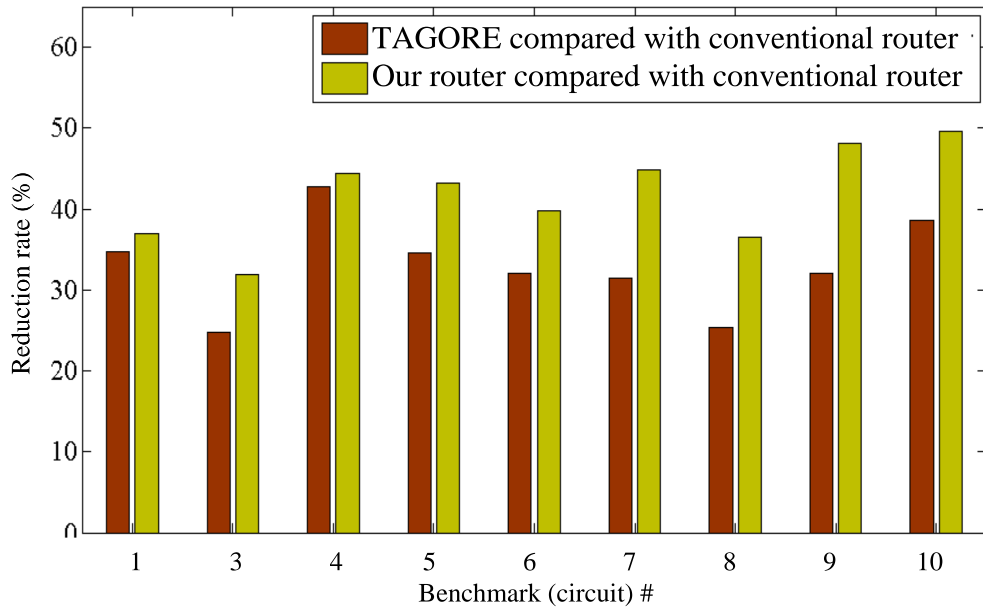


Figure 4.8. Reduction rates: comparison with a conventional router

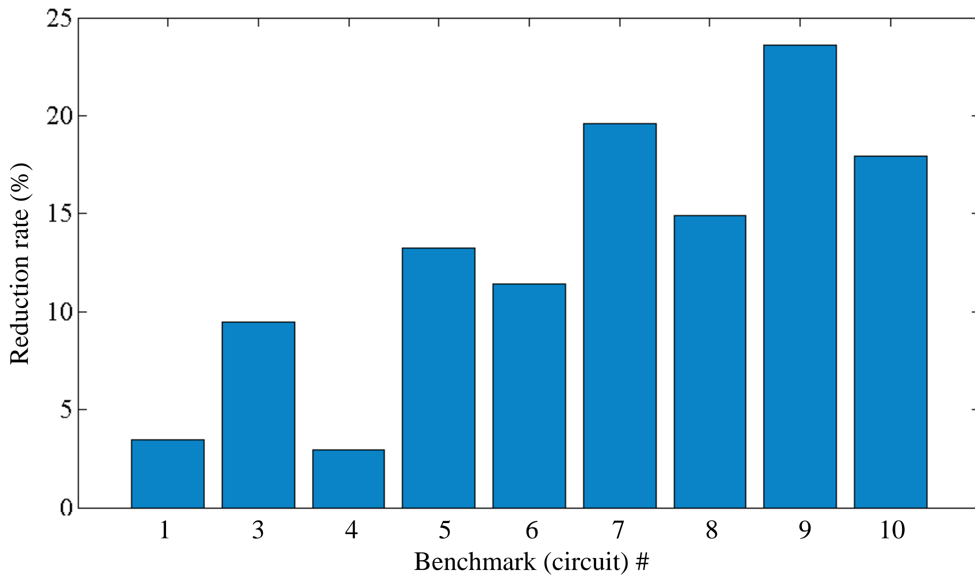


Figure 4.9. Trend in reduction rates: our router in comparison with TAGORE [58]

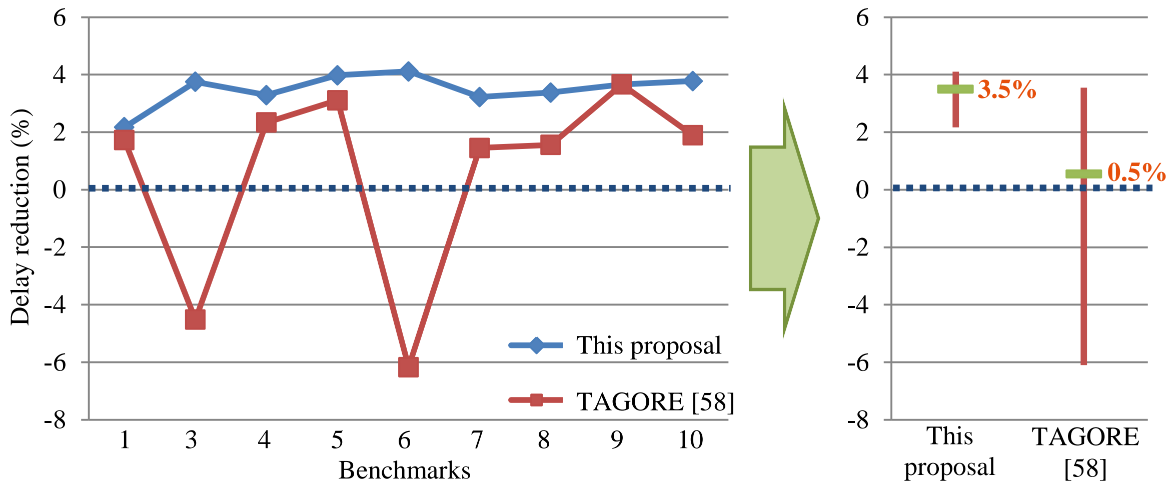


Figure 4.10. Delay reduction: comparison with a conventional router

compelling, our router gave better results than both TAGORE and a conventional router because our router uses temperature-aware delay as a major metric. In addition, TAGORE doesn't consider delay at all in selecting a path and just considers and compares maximum temperatures of two L-shaped paths; therefore, it increases delay when the thermal distribution on a selected path, of which maximum temperature is lower than the other, is constantly higher than the counterpart on the other path. In Figure 4.10, we can see our router consistently reduced the delay with the average reduction rate of 3.5%, while TAGORE increased the delay by up to 6%.

4.5. Summary

In this chapter, we presented a vision-inspired global router that increases the reliability of a chip by reducing the number of nets in hotspots while maintaining the delay as small as possible. Our router finds a limited number of nodes using a thermal profile of a chip and

image processing and computer vision techniques, and it expands the solution space efficiently and properly using the newly selected nodes. Our router is computationally efficient because it doesn't explore huge solution space, and it gives better results than previous L-shaped routers because it explores solution space that is expanded properly. In addition, it reduces delay of interconnects when compared with other routers because it uses temperature-aware delay of paths as a metric for routing.

CHAPTER 5. Conclusion

As we enter the deep submicron era, the number of transistors in a chip has increased at an alarming rate, and the power density distribution of a chip in current CMOS technology nodes is quite far from being uniform. For example, an SoC has multiple complex heterogeneous components on a chip, and different parts of a chip will have different power densities owing to different activity levels and different types of circuits on a die. As a result, thermal distribution of a chip also became non-uniform both temporally and spatially, and it causes a large number of problems such as reduced reliability, increased power consumption, and limited performance. In order to prevent these temperature-related issues, various DTM solutions have been proposed in recent years, and in order for those DTM solutions to work as intended, accurate temperature information on a full-chip scale needs to be provided in a timely manner.

RO-based thermal sensors are gaining popularity in sensing temperatures mainly due to their small form factor, low power consumption, and full digital CMOS compatibility. However, their reading accuracy is quite limited, and DTM solutions based on wrong temperature information from this type of sensors might work adversely and lead to catastrophic results in some extreme cases. As a way of increasing the reading accuracy of RO-based thermal sensors, we proposed a novel approach of using multiple virtual sensors, which were all generated from one physical sensor by adaptively switching its calibration points on the run. Simulation results show that the RMSE in temperature readings can be

reduced by up to 91.1% with the use of four virtual sensors in comparison with a single physical sensor case.

Secondly, we proposed methods of sensor allocation and full-chip thermal profile reconstruction. Thermal sensors were allocated with full-chip profile reconstruction in mind from the beginning, and we built a geometrical framework by analyzing the thermal profiles of a chip. Then we used the framework for sensor allocation and runtime full-chip profile reconstruction. Test results based on thousands of thermal profiles captured by a thermal imaging device show that the RMSE over an entire die area can be reduced by up to 36%, and the averaged absolute error at the hottest spot on a die can be reduced by up to 50% in comparison with a previous method when we use six thermal sensors.

In addition, a temperature-aware interconnect routing method was discussed. A basic assumption that the interconnect delay is solely dependent on the length of interconnects is not valid any more in deep submicron eras, and thermal effect on delay needs to be considered during interconnect design, especially for global interconnects. We proposed a method of global routing that considered the thermal distribution of substrates and its effect on interconnect delay so that the probability of chip failure due to interconnect failure could be minimized, and the performance degradation from increased delay could be prevented to some extent. We observed that the number of grid nodes in hotspots was reduced by up to 50 % when compared with the counterpart of a conventional router, while the delay of interconnects was reduced by up to 4.11%.

5.1. Practical design issues

In our proposals, we made several assumptions, and some of them might not be realistic in some cases. For that reason, it is necessary to review and adjust them to suit better each physical design case.

On the subject of sensor reading accuracy, we proposed a method regarding how to use multiple calibration points in an effective way to improve the accuracy. One thing to consider in choosing the proper number of calibration points or virtual sensors is the high cost for sensor calibration; thus, it is advisable to use a minimum number of virtual sensors as briefly discussed in chapter 1.

Our proposal on sensor placement is based on a basic assumption that it is desirable to limit the number of thermal sensors on a die even though thermal sensors are getting smaller nowadays. According to the recent design of RO-based thermal sensors in 65nm CMOS technology node [57], the die area consumed by one thermal sensor including all required components such as a counter and a voltage regulator was around 0.01mm^2, and it is still not small enough to be placed at any locations that we want for most cases, especially when we take the associated interconnect routings into account. Consequently, the aforementioned assumption might be valid for the next few technology generations to come. Another assumption that we made in this proposal is that the temperature readings of thermal sensors are accurate. When the accuracy in actual readings is degraded by several factors such as the low resolution of a sensor and its inherent inaccuracy, profile reconstruction results might have a lot larger errors than we expect. In order to deal with this practical issue appropriately, sensor selection needs to be performed with care. In our proposal, thermals sensors assigned

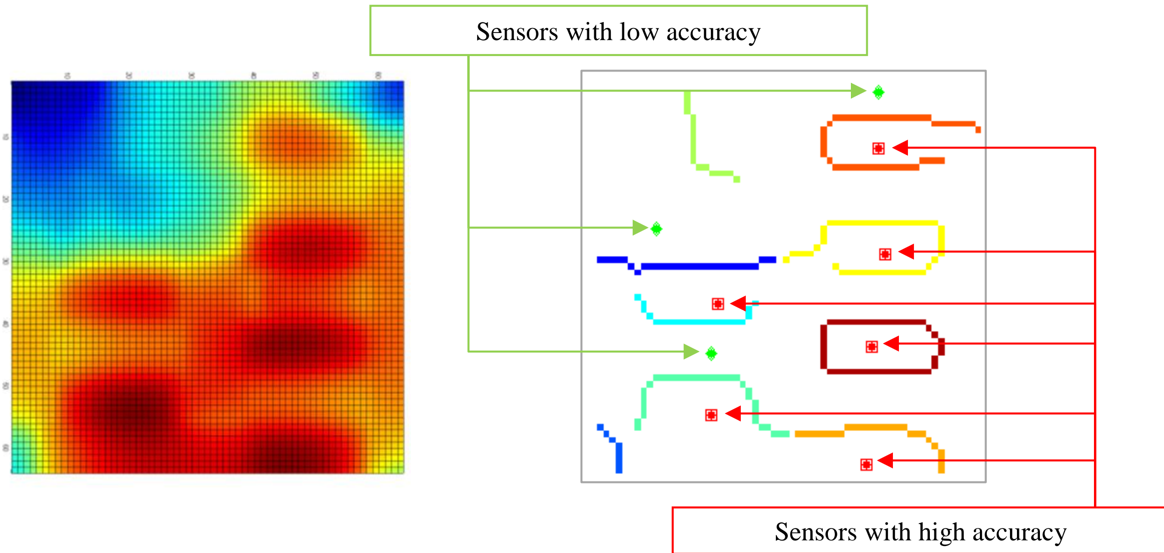


Figure 5.1. Practical design issues: sensor accuracy and sensor allocation

to the sensor points are required to have higher accuracy than the remaining sensors are; therefore, it would be logical to use thermal sensors with high reading accuracy and high resolution at the sensor point locations, while sensors with low accuracy are used at the remaining sensor candidate locations. If we apply the method proposed in chapter 2 for the improvement in reading accuracy, we might use four or more virtual sensors for each sensor at sensor point locations, and two virtual sensors for the remaining sensors. Again, high cost for calibration should be considered in choosing the proper number of virtual sensors.

As for the routing of global interconnects, we assumed congestion or densely-populated interconnects at some locations is not a big issue. In some locations where the congestion issue cannot be ignored, some modifications need to be made to the proposed method. Firstly, we can prioritize the nets of the circuits and then route important nets first. When the capacity of a certain location is filled up, all associated paths will be removed from the solution space so that those paths are not considered for the routing of the remaining nets. Secondly, we can

choose to route all the nets without any consideration on capacity first and then reroute some nets whose signal propagation delays have some slack. Thirdly, we can expand solution space for a location where congestion becomes an issue and reroute all the affected interconnects so that they can be spread out from the congested location using the expanded solution space.

5.2. Improvement of the proposals

For each proposed method, there still remain some chances of improvement. Regarding the proposed method to improve the accuracy of sensor readings, calibration points of a thermal sensor were uniformly distributed over the entire temperature range to be monitored. If we can acquire the information on the temperature variations at the spot where the sensor is located or if we have thermal profiles of a chip, we can make better choices regarding the distribution of calibration points of each virtual sensor using the information as explained in Figure 5.2. We can keep the distance between adjacent calibration points short when the entire temperature range to be monitored is not wide. We can also adjust the distance between adjacent calibration points based on the importance of the temperature range to be monitored. For example, we can use a long distance between calibration points over a temperature range whose accuracy is less important and use a short distance over a temperature range whose accuracy is important.

In our proposal concerning sensor allocation and thermal profile reconstruction, we used DCT for the transform-based reconstruction. Considering the fact that DCT basis is a general basis that works equally well on various kinds of signals, there remains a great opportunity to

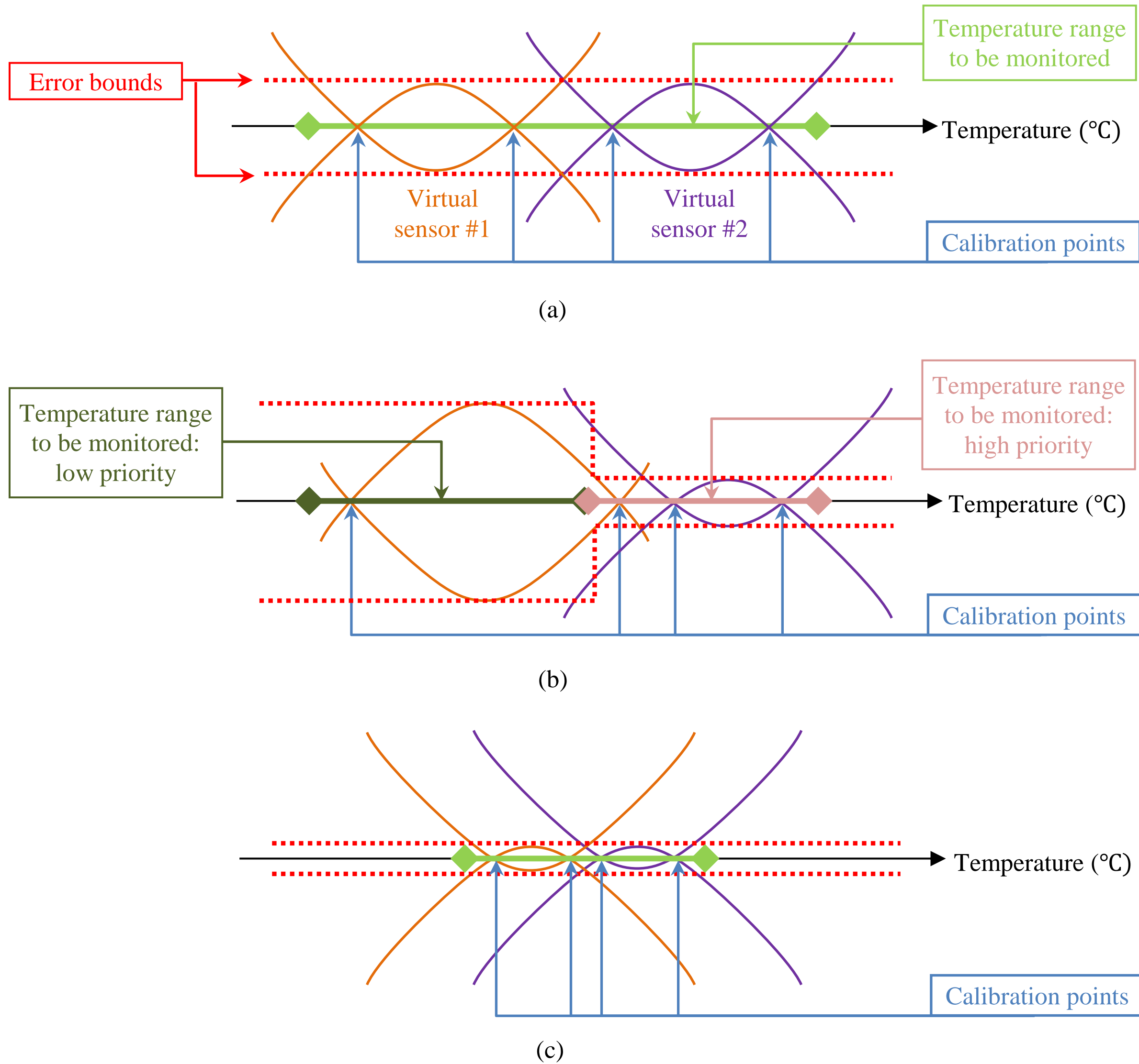


Figure 5.2. Calibration points of virtual sensors: (a) uniform error bounds over the wide temperature range, (b) non-uniform error bounds over the wide temperature range, (c) uniform error bounds over the narrow temperature range

improve the accuracy of profile reconstruction by using an optimal transform basis specifically developed for the thermal profiles of a given chip. Additional improvement can be made by generating a reference thermal profile in a smarter way. In the proposal, we used averaging to generate a reference thermal profile. One possible issue of this approach, i.e.,

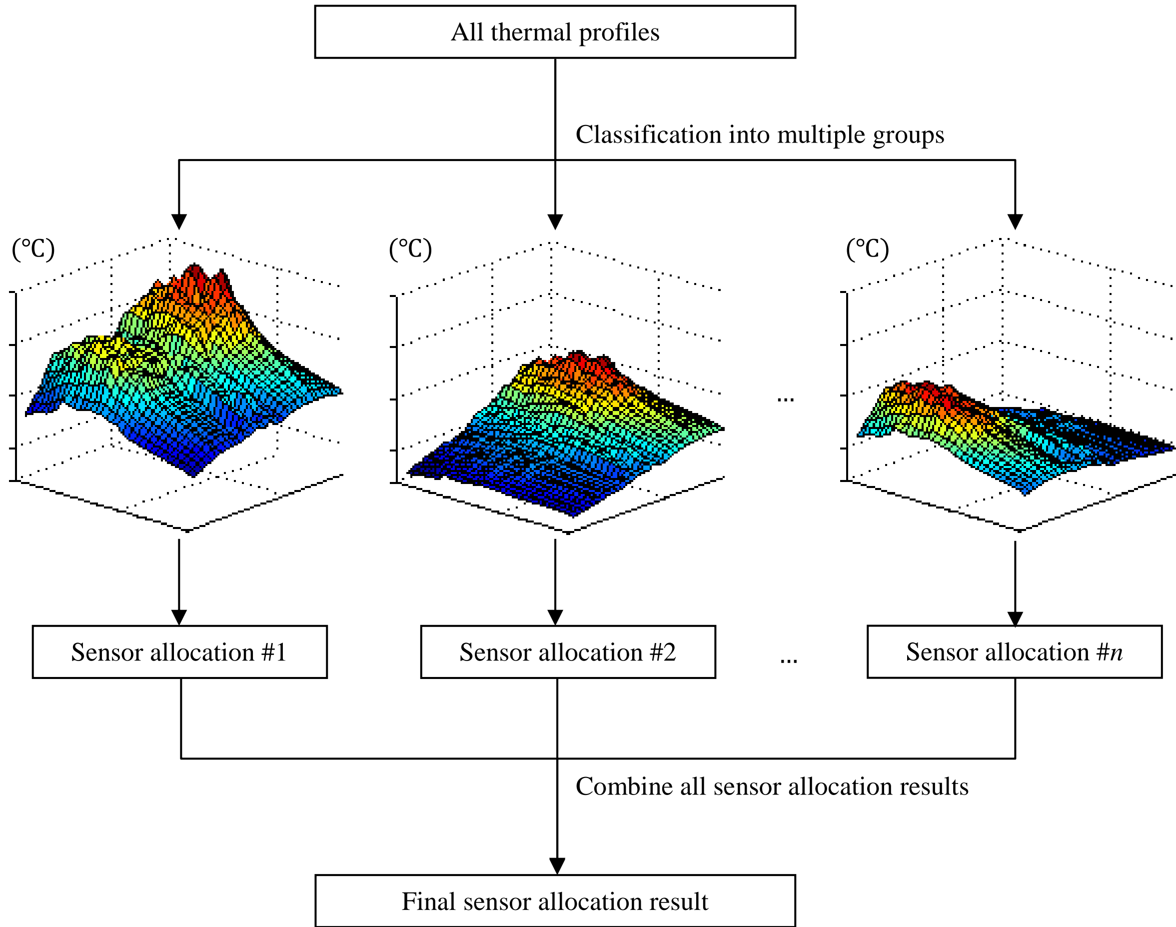


Figure 5.3. Improvement of the proposals: sensor allocation

simply averaging all thermal profiles, is that some unique properties of minor thermal profiles, which appear intermittently, can be cancelled out or ignored during the averaging process. This can be also an issue of major thermal profiles in some cases. In order to prevent this issue, we can first classify entire thermal profiles into several distinctive profile groups using pattern classification techniques such as Principal Components Analysis (PCA) [81], and then we apply the proposed sensor allocation method to each group separately. Sensor allocation result of each and every group can be combined to generate a final set of thermal sensors. If we need to limit the total number of thermal sensors, we can remove some thermal sensors

that are associated with the groups that appear sporadically. This might increase the total number of sensors to be allocated, but we can achieve a lot better reconstruction results because we assign thermal sensors so that even minor thermal profiles can be reconstructed equally well.

5.3. Directions for future research

Clock tree synthesis is another topic that requires temperature-aware design. As explained in chapter 4, interconnect delay depends on the thermal distribution of substrates. As a result, clock tree synthesis methods built on an assumption that the thermal distribution of a chip across a die is uniform should be modified accordingly in order to prevent any clock skew related issues. Recently, several approaches [16] [82] [83] were proposed to minimize the clock skew variation caused by temporal and spatial temperature variation. In the future, we can extend our work presented in chapter 4 to improve those methods by allowing the actual delay of each path to be calculated and more accurate clock skew to be considered in path selection.

The advent of 3D integrated circuits also gave particular prominence to temperature-aware design and DTM solutions. 3D ICs [84] are usually made by stacking up 2D planar IC structures, and it allows us to put a lot more transistors on the exactly same foot print. In addition, the interconnect length can be drastically reduced by the use of short vertical interconnections between tiers, which are usually named Through Silicon Vias or TSVs in short; thus, we can reduce power consumption caused by long global interconnects and also can solve delay-induced issues using TSVs. One of the reasons why we need to pay a lot more

attention to the thermal issues of 3D ICs is that cooling solutions for 3D ICs are not appreciably different from the ones used for conventional 2D ICs. Increase in power density due to stacking is another reason why temperature-aware design is indispensable in 3D ICs. All proposed methods in this work were designed mainly for 2D planar ICs, and they might not work as intended in case of 3D ICs because we did not consider 3D IC specific issues such as three dimensional thermal profiles instead of planar thermal profiles, more complicated relationship between the temperature of substrates and the delay of interconnects, etc. As a result, it might be necessary to update them or simply propose all new methods that are suitable for 3D ICs in the near future.

In this work, a part of low-level temperature-aware design methodologies were explored; how to prepare accurate temperature information on a full-chip scale so that DTM solutions can use the information and manage resources in an efficient way to improve the thermal situation of a chip; how to route global interconnects so that delay-related issues can be reduced, and the severity or damage can be alleviated even in extreme thermal conditions. Temperature-aware design does not provide complete protection against severe thermal conditions all by itself; it needs to be accompanied by proper DTM solutions so that they work together as a whole system to provide a reliable working environment. Naturally, we can propose a new fine-grain DTM technique as a next step that is optimized for a chip based on the extensive knowledge about the chip; temperature-aware design methods used for the chip; the characteristics of each component of the chip such as the location of thermal sensors and their types and accuracy; methods used for full-chip thermal profile reconstruction, etc.

As we discussed in this work, there are quite strong correlations among the performance, power, reliability, and temperature of a chip or a system. Engineers and researchers have tried to improve the performance of a chip or a system for decades, and a large number of techniques or solutions have been proposed to deal with the associated issues concerning the temperature, power consumption, and reliability of high performance chips. However, the most trustworthy and safe choice was always limiting the performance of chips in order to reduce the temperature and power consumption and also to increase the reliability. That means, there still remain a great number of questions to be answered clearly in order to design a chip or a system with great performance and reliability while consuming less power and generating less heat, and the on-going research on temperature-aware design and DTM solutions might be able to provide some insight or some good answers to those questions.

REFERENCES

[1] S. Borkar, "Thousand Core Chips - A Technology Perspective," in *DAC*, 2007.

[2] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," in *ISCA*, 2006.

[3] T. Sato, J. Ichimiya, N. Ono, K. Hachiya and M. Hashimoto, "On-chip Thermal Gradient Analysis and Temperature Flattening for SoC Design," *IEICE Trans. Fundamentals,* Vols. E88-A, no. 12, pp. 3382-3389, Dec. 2005.

[4] P. Gronowski, W. Bowhill, R. Preston, M. Gowan and R. Allmon, "High-Performance Microprocessor Design," *IEEE J. Solid-State Circuits,* vol. 33, pp. 676-686, 1998.

[5] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter Variations and Impact on Circuits and Microarchitecture," in *DAC*, 2003.

[6] J. Black, "Electromigration - A Brief Survey and Some Recent Results," *IEEE Trans. Electron Devices,* vol. 4, 1969.

[7] K. Skadron, M. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy and D. Tarjan, "Temperature-Aware Microarchitecture: Modeling and Implementation," *ACM Trans. Architecture and Code Optimization,* vol. 1, pp. 94-125, 2004.

[8] J. Blair, P. Ghate and C. Haywood, "Concerning Electromigration in Thin Films," *Proc. IEEE lett.,* vol. 59, pp. 1023-1024, 1971.

[9] R. d. Orio, H. Ceric and S. Selberherr, "Physically Based Models of Electromigration: from Black's Equation to Modern TCAD Models," *Microelectronics Reliability,* vol. 50,

no. 6, pp. 775-789, 2010.

[10] J. R. Lloyd, "Electromigration Failure," *J. Applied Physics,* vol. 69, pp. 7601-7604, 1991.

[11] M. Shatzkes and J. R. Lloyd, "A Model for Conductor Failure Considering Diffusion Concurrently with Electromigration Resulting in A Current Exponent of 2," *J. Applied Physics,* vol. 59, pp. 3890-3893, 1986.

[12] J. Lienig, "Electromigration and Its Impact on Physical Design in Future Technologies," in *ISPD*, 2013.

[13] S. Im and K. Banerjee, "Full chip thermal analysis of planar (2-D) and vertically integrated (3-D) high performance ICs," in *IEEE Int. Electron Devices Meeting*, 2000.

[14] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors (ITRS): Interconnect," 2007.

[15] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors (ITRS): interconnect," 2011.

[16] A. Ajami, K. Banerjee and M. Pedram, "Modeling and Analysis of Nonuniform Substrate Temperature Effects on Global ULSI Interconnects," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,* vol. 24, pp. 849-861, 2005.

[17] W. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *J. Applied Physics,* vol. 19, pp. 55-63, 1948.

[18] A. Ajami, K. Banerjee and M. Pedram, "Analysis of Non-Uniform Temperature-Dependent Interconnect Performance in High Performance ICs," in *DAC*, Las Vegas, NV USA, 2001.

[19] E. Morifuji, T. Yoshida, M. Kanda, S. Matsuda, S. Yamada and F. Matsuoka, "Supply and Threshold-Voltage Trends for Scaled Logic and SRAM MOSFETs," *IEEE Trans. Electron Device,* pp. 1427-1432, 2006.

[20] A. Chandrakasan, W. Bowhill and F. Fox, Design of High-Performance Microprocessor Circuits, IEEE Press, 2001.

[21] K. Roy and S. C. Prasad, Low Power CMOS VLSI Circuit Design, John Wiley & Sons, Inc., 2000.

[22] A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy and C. Kim, "Leakage Power Analysis and Reduction for Nanoscale Circuits," *IEEE Micro,* vol. 26, pp. 68-80, 2006.

[23] K. Mistry et al., "A 45nm Logic Technology with High-k+Metal Gate Transistors, Strained Silicon, 9 Cu Interconnect Layers, 193nm Dry Patterning, and 100% Pb-free Packaging," in *IEEE Int. Electron Devices Meeting*, 2007.

[24] S. Gunther, F. Binns, D.M.Carmean and J. Hall, "Managing the Impact of Increasing Mircoprocessor Power Consumption," *Intel Technology Journal,* vol. 5, no. 1, pp. 1-9, 2001.

[25] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," in *HPCA*, 2011.

[26] A. Kumar, L. Shang, L. Peh and N. K. Jha, "System-level Dynamic Thermal Management for High-performance Microprocessors," *IEEE TCAD,* 2008.

[27] R. Shelar and M. Patyra, "Impact of Local Interconnects on Timing and Power in a High Performance Microprocessor," in *ISPD*, 2010.

[28] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, "Temperature-Aware Computer Systems: Opportunities and Challenges," in *IEEE MICRO*, 2003.

[29] J. Chabloz and A. Hemani, "Distributed DVFS Using Rationally-Related Frequencies and Discrete Voltage Levels," in *ISLPED*, 2010.

[30] M. Gerards, J. L. Hurink and J. Kuper, "On the Interplay between Global DVFS and Scheduling Tasks with Precedence Constraints," *IEEE Trans. Comput.,* 2014.

[31] S. Herbert and D. Marculescu, "Analysis of Dynamic Voltage/Frequency Scaling in Chip-Multiprocessors," in *ISLPED*, 2007.

[32] A. Baniasadi and A. Moshovos, "Instruction Flow-Based Front End Throttling for Power-Aware High Performance Processors," in *ISLPED*, 2001.

[33] E. Rohou and M. Smith, "Dynamically Managing Processor Temperature and Power," in *Workshop on Feedback Directed Optimization*, 1999.

[34] J. Donald and M. Martonosi, "Leveraging Simultaneous Multithreading for Adaptive Thermal Control," in *2nd workshop on Temperature-Aware Computer Systems*, 2005.

[35] K. J. Lee and K. Skadron, "Using Performance Counters for Runtime Temperature Sensing in High-Performance Processors," in *Workshop on High-Performance, Power-Aware Computing*, 2005.

[36] A. K. Coskun, T. S. Rosing and K. Whisnant, "Temperature Aware Task Scheduling in MPSoCs," in *DATE*, 2007.

[37] J. Long, S. Memik, G. Memik and R. Mukherjee, "Thermal Monitoring Mechanisms for

Chip Multiprocessors," *ACM Trans. Architecture and Code Optimization,* vol. 5, 2008.

[38] H. Sanchez, R. Philip, J. Alvarez and G. Gerosa, "A CMOS Temperature Sensor for PowerPC RISC Microprocessors," in *Symp. VLSI Circuits*, 1997.

[39] Intel, "Intel® Pentium® III Processor Thermal Design Guide: application note," 2001. [Online]. Available: http://download.intel.com/design/intarch/applnots/27332504.pdf.

[40] S. O. Memik, R. Mukherjee, M. Ni and J. Long, "Optimizing Thermal Sensor Allocation for Microprocessors," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.,* vol. 27, pp. 516-527, 2008.

[41] SPEC-CPU2000, "Standard performance evaluation council, performance evaluation in the new millennium, v.1.1," 2000.

[42] R. Mukherjee, S. Mondal and S. Memik, "Thermal Sensor Allocation and Placement for Reconfigurable Systems," in *ICCAD*, 2006.

[43] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *5th Berkeley Symp. Mathematical Statistics and Probability*, 1967.

[44] R. Mukherjee and S. O. Memik, "Systematic Temperature Sensor Allocation and Placement for Microprocessors," in *DAC*, 2006.

[45] R. Rao, S. Vrudhula and C. Chakrabarti, "Throughput of Multi-core Processors Under Thermal Constraints," in *ISLPED*, 2007.

[46] J. K. John, J. S. Hu and S. G. Ziavras, "Optimizing the Thermal Behavior of Subarrayed Data Caches," in *ICCD*, 2005.

[47] S. Kaxiras, Z. Ju and M. Martonosi, "Cache Decay: Exploiting Generational Behavior to

Reduce Cache Leakage Power," in *ISCA*, 2001.

[48] A. Nowroz, R. Cochran and S. Reda, "Thermal Monitoring of Real Processors: Techniques for Sensor Allocation and Full Characterization," in *DAC*, 2010.

[49] A. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989.

[50] Scalable Computing Systems Lab., [Online]. Available: http://scale.engin.brown.edu/tools/.

[51] Standard Performance Evaluation Corporation, "SPEC CPU 2006," [Online]. Available: http://www.spec.org/cpu2006/index.html.

[52] A. V. Oppenheim, A. S. Willsky and S. H. Nawab, Signals and Systems 2nd Ed., Prentice Hall, 1997.

[53] Y. Zhang, B. Shi and A. Srivastava, "A Statistical Framework for Designing On-chip Thermal Sensing Infrastructure in Nano-scale Systems," in *ISPD*, 2010.

[54] W. Wu, L. Jin, J. Yang, P. Liu and S. Tan, "A Systematic Method for Functional Unit Power Estimation in Mircoprocessors," in *DAC*, 2006.

[55] M. D. Powell, A. Biswas, J. S. Emer, S. Mukherjee, B. Sheikh and S. Yardi, "CAMP: A Technique to Estimate Per-Structure Power at Run-time using a Few Simple Parameters," in *HPCA*, 2009.

[56] H. Wang, S. Tan, S. Swarup and X. Liu, "A Power-Driven Thermal Sensor Placement Algorithm for Dynamic Thermal Management," in *DATE*, 2013.

[57] N. Testi and Y. Xu, "A 0.2nJ/sample 0.01mm2 Ring Oscillator Based Temperature Sensor for On-Chip Thermal Management," in *ISQED*, 2013.

[58] A. Gupta, N. Dutt, F. Kurdahi, K. Khouri and M. Abadir, "Thermal Aware Global Routing of VLSI Chips for Enhanced Reliability," in *ISQED*, 2008.

[59] Y. Zhang and A. Srivastava, "Accurate Temperature Estimation Using Noisy Thermal Sensors for Gaussian and non-Gaussian cases," *IEEE TVLSI,* 2011.

[60] E. Rotem, J. Hermerding, C. Aviad and C. Harel, "Temperature Measurement in the Intel® CoreTM Duo Processor," in *THERMINIC*, 2006.

[61] H. Lakdawala, Y. W. Li, A. Raychowdhury, G. Taylor and K. Soumyanath, "A 1.05V 1.6mV 0.45 oC 3σ-Resolution Δ∑-based Temperature Sensor with Parasitic-resistance Compensation in 32nm CMOS," in *IEEE ISSCC*, 2009.

[62] K. Souri, Y. Chae and K. A. A. Makinwa, "A CMOS Temperature Sensor with a Voltage Calibrated Inaccuracy of ±0.15 ℃ (3sigma) from -55 ℃ to 125 ℃," in *IEEE ISSCC*, 2012.

[63] Y. S. Lin, D. Sylvester and D. Blaauw, "An Ultra Low Power 1V, 220nW Temperature Sensor for Passive Wireless Applications," in *IEEE CICC*, 2008.

[64] Y. Zhang and A. Srivastava, "Accurate Temperature Estimation Using Noisy Thermal Sensors," in *DAC*, 2009.

[65] S. Lu, R. Tessier and W. Burleson, "Collaborative Calibration of On-Chip Thermal Sensors Using Performance Counters," in *ICCAD*, 2012.

[66] S. Sharifi and T. S. Rosing, "Accurate Direct and Indirect On-Chip Temperature Sensing for Efficient Dynamic Thermal Management," *IEEE TCAD,* 2010.

[67] A. S. Sedra and K. C. Smith, Microelectronic Circuits 5th edition, Oxford University

Press, 2004.

[68] S. Sze, Physics of Semiconductor Devices 2nd edition, New York: John Wiley and Sons, 1981.

[69] IBM, "Using thermal diodes in the PowerPC970MP Processor," IBM white paper, 2006.

[70] J. Choi, C. Cher, H. Franke, H. Hamann, A. Weger and P. Bose, "Thermal-aware Task Scheduling at the System Software Level," in *ISLPED*, 2007.

[71] S. Nassif, "Delay Variability: Sources, Impact and Trends," in *IEEE ISSCC*, 2000.

[72] W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45nm Design Exploration," in *ISQED*, 2006.

[73] J. Canny, "A Computational Approach to Edge Detection," *IEEE TPAMI,* 1986.

[74] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, 1984.

[75] J. Leader, Numerical Analysis and Scientific Computation, Pearson, 2004.

[76] W. Huang, S. Ghosh, S. Velusamy, K. Skadron, K. Sankaranarayanan and M. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early-stage VLSI Design," *IEEE TVLSI,* vol. 14, pp. 501-513, 2006.

[77] N. Sherwani, Algorithms for VLSI Physical Design Automation, Kluwer Academic Publisher, 1999.

[78] K. Lu and D. Pan, "Reliability-aware Global Routing under Thermal Considerations," in *ASQED*, 2009.

[79] T. H. Cormen, . E. Leiserson and . L. Rivest, Introduction to Algorithms, 3rd edition, MIT Press, 2009.

[80] Kastner Research Group, [Online]. Available: http://cseweb.ucsd.edu/~kastner/labyrinth_vault/benchmarks/index.html.

[81] A. K. Jain, R. Duin and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE TPAMI,* vol. 22, pp. 4-37, 2000.

[82] M. Cho, S. Ahmed and D. Z. Pan, "TACO: Temperature Aware Clock-tree Optimization," in *ICCAD*, 2005.

[83] C. Liu, J. Su and Y. Shi, "Temperature-Aware Clock Tree Synthesis Considering Spatiotemporal Hot Spot Correlations," in *ICCD*, 2008.

[84] A. Coskun, J. Ayala, D. Atinza, T. Rosing and Y. Leblebici, "Dynamic Thermal Management in 3D Multicore Architectures," in *DATE*, 2009.

[85] B. Meyerson, in *Semico Impact Conference*, Taiwan, 2004.

[86] Y. Zhang, Y. Li, X. Li and S. Yao, "Strip-and-Zone Micro-Channel Liquid Cooling of Integrated Circuits Chips With Non-Uniform Power Distributions," in *ASME 2013 Heat Transfer Summer Conference*, 2013.

[87] S. Reda, R. J. Cochran and A. Nowroz, "Improved Thermal Tracking for Processors Using Hard and Soft Sensor Allocation Techniques," *IEEE Trans. Comput.,* vol. 60, pp. 841-851, 2011.

[88] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors (ITRS) - System Driver 2010 update," 2010.