國立交通大學

資訊科學與工程研究所

碩 士 論 文

微 處 理 器 核 心 熱 分 布 量 測 與 模 擬 平 台 之 建 構

The Construction of Thermal Measurement and Simulation Platforms for Microprocessors

研 究 生：王星斐

指導教授：曹孝櫟　教授

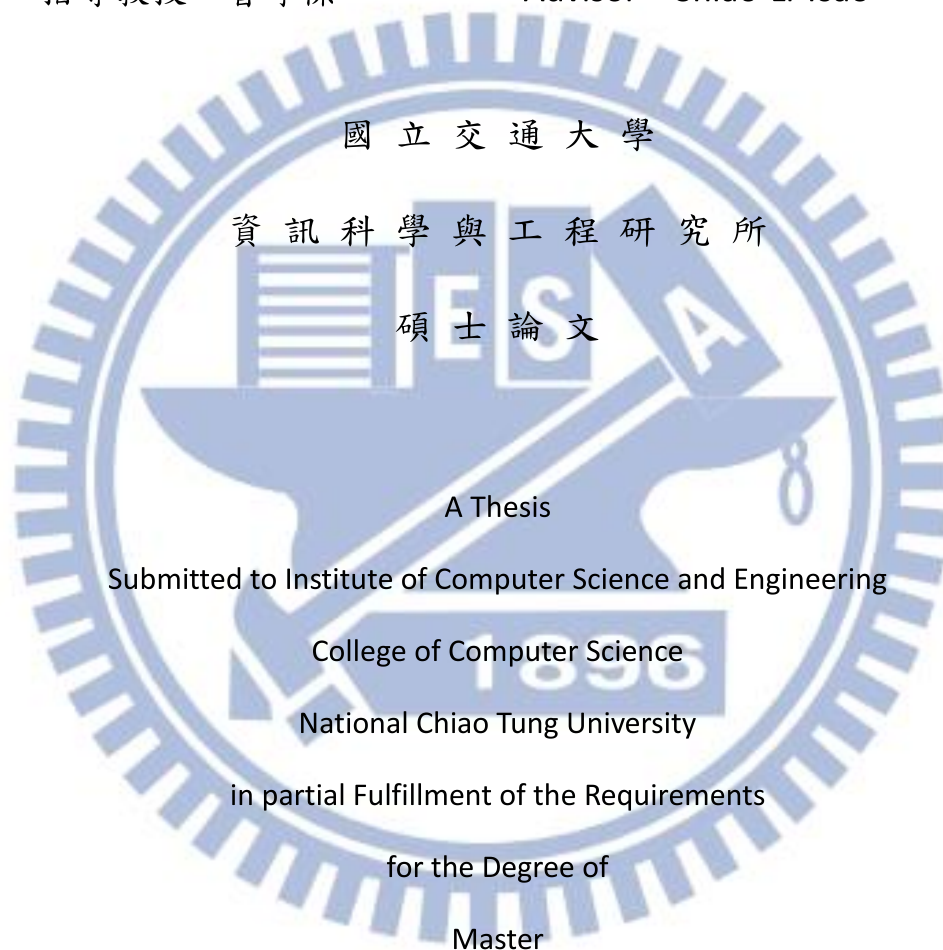
中 華 民 國 一 百 零 三 年 十一 月

微處理器核心熱分布量測與模擬平台之建構

The Construction of Thermal Measurement and Simulation Platforms for Microprocessors

研 究 生：王星斐　　　　Student：Hsing-Fei Wang

指導教授：曹孝櫟　　　　Advisor：Shiao-Li Tsao



國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

November 2014

Hsinchu, Taiwan, Republic of China

中華民國一百零三年十一月

微處理器核心熱分布量測與模擬平台之建構

研究生：王星斐　　　　　　　　　　　　指導教授：曹孝櫟 博士

國立交通大學
資訊科學與工程研究所
新竹市大學路1001號

碩士論文

摘要

隨著先進處理器的發展，工作頻率和功耗快速上升到臨界點，隨之所帶來大量的熱能損耗，也讓處理器的工作頻率和表現受限在此，因此需要散熱的技術來讓處理器可以提升效能與可靠性。 也因為先進製程技術的精進，熱能損耗帶來更多的靜態耗電，耗電又會造成熱能的損耗，兩者相互影響加劇，所以為了要有效控制熱能損耗，熱能管理是現在重要的研究課題之一。 我們可以透過熱能管理來控制運作溫度，並有效提升處理器效能與可靠性，也可以降低熱損耗所帶來的靜態耗電。 為了要有效發展和使用熱能管理，獲得微處理器內部各個位置上精確的溫度資訊則是熱能管理中重要的第一步。 在本論文中，我們結合並創建模擬平台，可以在執行程式時，透過獲取核心中各個功能單元的效能計數器，來得到核心的溫度分布，並可以使用所得到的各功能單元的溫度，來計算因為熱能損耗所帶來的靜態耗電。 也搭配使用熱像儀，在實際的平台上，建立動態的熱能評估。

The Construction of Thermal Measurement and Simulation Platforms for Microprocessors

Student: Hsing-Fei Wang   Advisor: Shiao-Li Tsao
Institutes of Computer Science and Engineering
National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300, ROC

Abstract

Modern CPUs with increasing core frequency and power are rapidly reaching a point where the CPU frequency and performance are limited by the amount of heat, then it need cooling technology to enhance performance and reliability. Because of the enhancement of the technology size, heat dissipation also result in more leakage power consumption and the increasing power will generate more heat then they will influence with each other heavily. Therefore, in order to decrease the temperature of the processors, it is the most important part about how to use Dynamic Thermal Management [24] to control the temperature in recent study topic. We can increase the performance and reliability of processors through efficiently using DTM to manage the operating temperature. To obtain detail thermal distribution of a microprocessor is one of critical tasks for thermal management which improves the reliability, performance, etc. of modern microprocessors significantly. In this thesis, we combine and construct the simulation platform to derive detailed temperature information of a microprocessor by catching the performance counters in each functional unit of the core when we run a program and we would use detailed temperature information to calculate the leakage power which generated from heat. At last, we will construct real time thermal estimation on the real platform with using the infrared camera.

誌謝

俗話說得好，不經一番寒徹骨 焉得梅花撲鼻香，經過總總的挑戰和磨練，終於順利完成本文。 在這條漫長的研究道路上，要感謝一路上大家對我的支持，和提攜。 首先，我想感謝我的指導老師，曹孝櫟教授，在研究的道路上，老師總是一步一步的引導，勾起我們對研究的熱忱，並將繁雜的理論知識用深入淺出，言簡意賅的方式，讓我們裡解其中的道理和精隨。一路上，從老師身上學到的東西多不勝數，不只在知識學問上滿載而歸，更了解做學問該有的態度，非常感謝老師這幾年的提拔和教導。 同時，我也要感謝我的口試委員，黃育綸教授、賴伯承教授，感謝你們針對我的論文和研究上，給予珍貴的建議，讓我的論文可以更加完備。

接下來，我想感謝在我求學的階段上，不斷給我支持的家人們，他們辛辛苦苦地提供我良好的學習環境，並讓我開開心心的長大，毫無擔憂的全力去學習，讓我有這個機會能順利完成碩士學位。

最後，我想感謝實驗室的所有夥伴，特別感謝培書學長、承威學長、勇旗學長、建明學長，他們是實驗室的大前輩、引路者，他們在我的研究道路上不只給予很多寶貴的研究知識，更將求學經驗和認真的學習態度毫不保留的傳承下來，我也一天天的更加茁壯並學會了很多實用的專業技能，尤其感謝培書學長總是陪我奮鬥到最後一刻，能和我一起討論、研究，並解決我心中很多的疑惑和瓶頸，一路上就像一盞明燈，讓我能在研究的道路上走得更加穩定和順暢。 也非常感謝同屆一起奮鬥的同袍宇安、佳駿、時粹、冠志，我們一起進到實驗室，一起奮鬥，一起追尋未知的事物，一起經歷種種，你們豐富了我的研究生活，讓我的生活五彩繽紛。也要感謝實驗室的學弟國瑋、子陽、學文、晉緯、Arthur 和濟韋，幫忙處理很多實驗室事務，因為你們的陪伴，我在實驗室更覺得開心，讓我在研究上更加得心應手。 最後，再次感謝一路給我幫助或打氣過的人，謝謝。

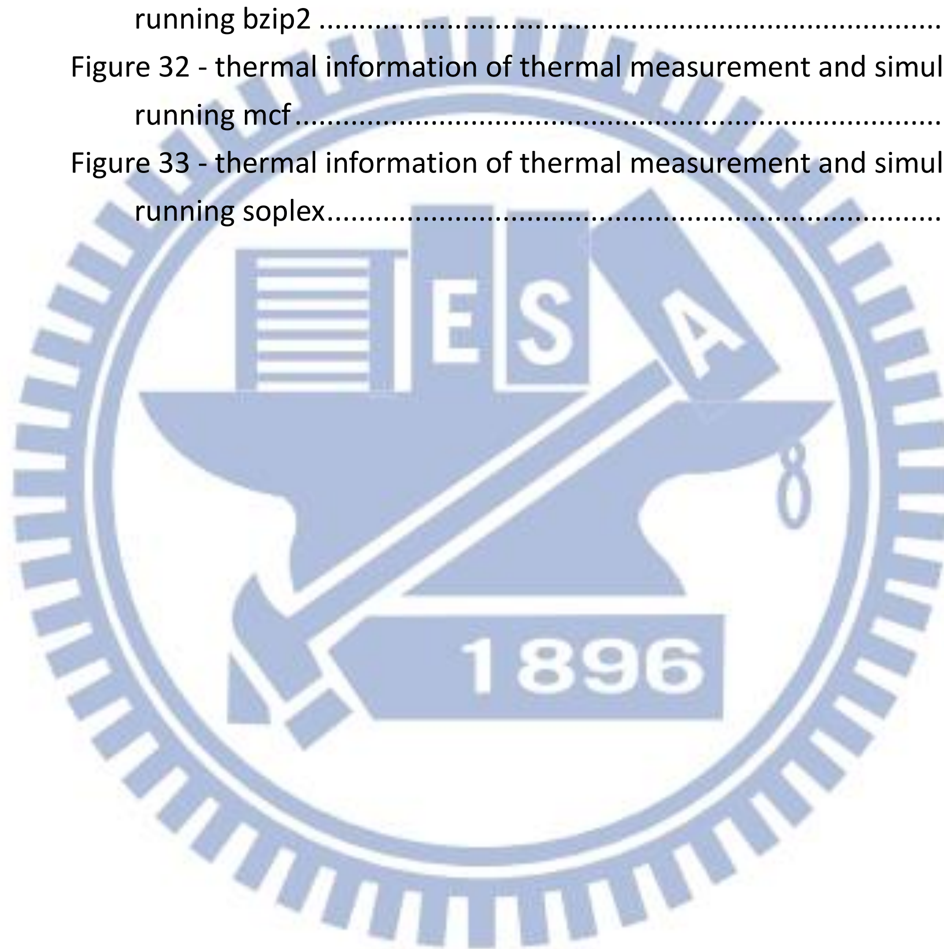
Table of Contents

摘要 i
Abstract ii
誌謝 iii
Table of Contents iv
List of Figures v
List of Tables vii
List of Equation viii
I. Introduction 1
II. Related Works 3
III. Simulation Platform for Microprocessors 5
    3.1 Architecture of Simulation Platform 6
    3.2 Thermal Estimation under Different Technology Size Using Different Compiler Optimization 8
IV. Counter-Based Approach 18
    4.1 Verification 19
V. Thermal Simulation and Measurement on Real Platform 21
    5.1 Thermal Simulation on Real Platform 22
    5.2 Thermal Measurement on Real Platform 23
VI. Experimental Setup 23
    6.1 Floor Plan of Core 23
    6.2 Infrared Camera 25
    6.3 Counter-Based power model on real platform 26
VII. Experimental Results 27
VIII. Conclusions and Future Works 30
Reference 31

List of Figures

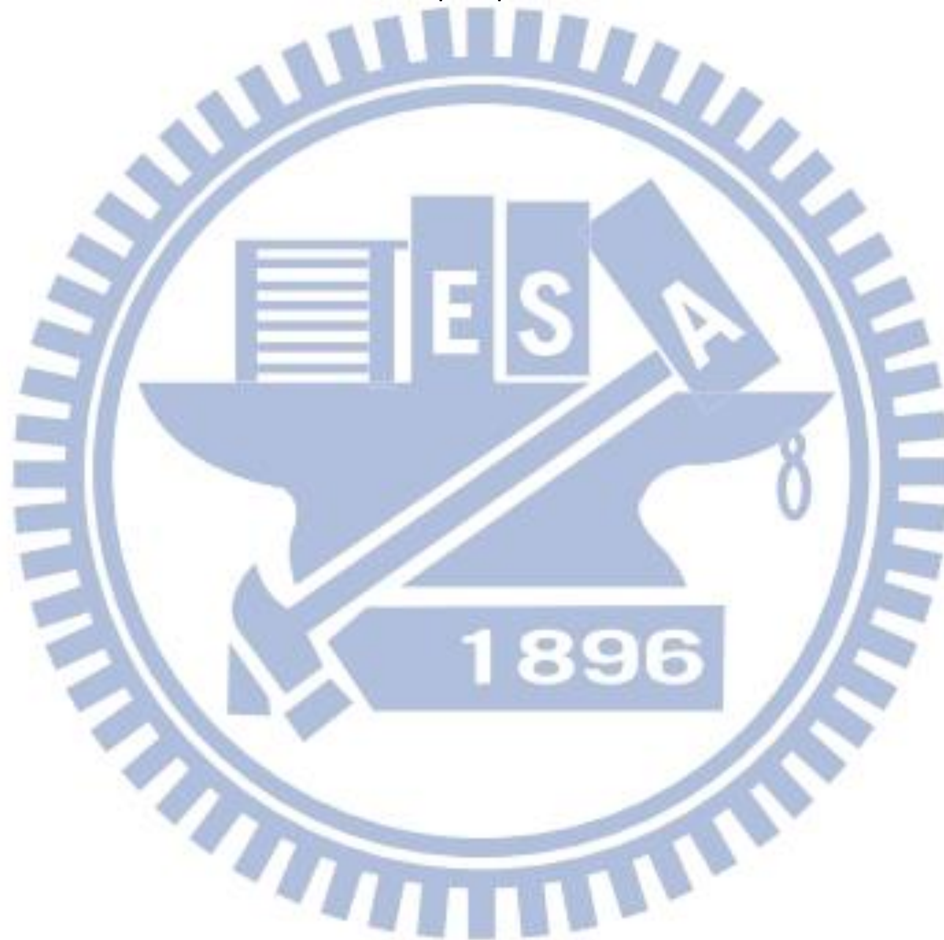
Figure 1 - feedback loop exists between temperature and power 2
Figure 2 - classification of thermal estimation platform...............................4
Figure 3 – flow of simulation platform 8
Figure 4 - ALPHA 21264 processor floor plan 9
Figure 5 – thermal information of functional units when run Bzip2 using O1 and O2 under 70 nm 11
Figure 6 - thermal information of functional units when run Bzip2 using O1 and O2 under 22 nm 11
Figure 7 – Execution time under 70 nm and 22 nm using O1 and O2 respectively 12
Figure 8 – dynamic power under 70 nm and 22 nm using O1 and O2 respectively 13
Figure 9 - dynamic energy under 70 nm and 22 nm using O1 and O2 respectively 13
Figure 10 – proportion of dynamic power and leakage power under 70 nm and 22 nm using O1 and O2 respectively 14
Figure 11 – leakage power under 70 nm and 22 nm using O1 and O2 respectively 15
Figure 12 - leakage energy under 70 nm and 22 nm using O1 and O2 respectively 15
Figure 13 – total power under 70 nm and 22 nm using O1 and O2 respectively 16
Figure 14 - total energy under 70 nm and 22 nm using O1 and O2 respectively 17
Figure 15 – energy delay product under 70 nm and 22 nm using O1 and O2 respectively 18
Figure 16 – the difference of EDP between O1 and O2 under 70 nm and 22 nm 18
Figure 17 – simulation-based and counter-based platform 19
Figure 18 – thermal information of functional units when run program If .20
Figure 19 - thermal information of functional units when run program Iffs 21
Figure 20 – flow of thermal simulation on real platform............................22
Figure 21 – architecture of thermal simulation on real platform 22
Figure 22 - flow of thermal measurement on real platform........................23

Figure 23 – floor plan of the chip 24
Figure 24 – floor plan of the core 24
Figure 25 – entire floor plan of chip 25
Figure 26 – FLIR SC7000 25
Figure 27 – environment setup for FLIR SC7000 26
Figure 28 - capture current power of chip 27
Figure 29 – thermal visualizer of thermal measurement 28
Figure 30 – thermal visualizer of thermal simulation 28
Figure 31 – thermal information of thermal measurement and simulation running bzip2 29
Figure 32 - thermal information of thermal measurement and simulation running mcf 29
Figure 33 - thermal information of thermal measurement and simulation running soplex 30



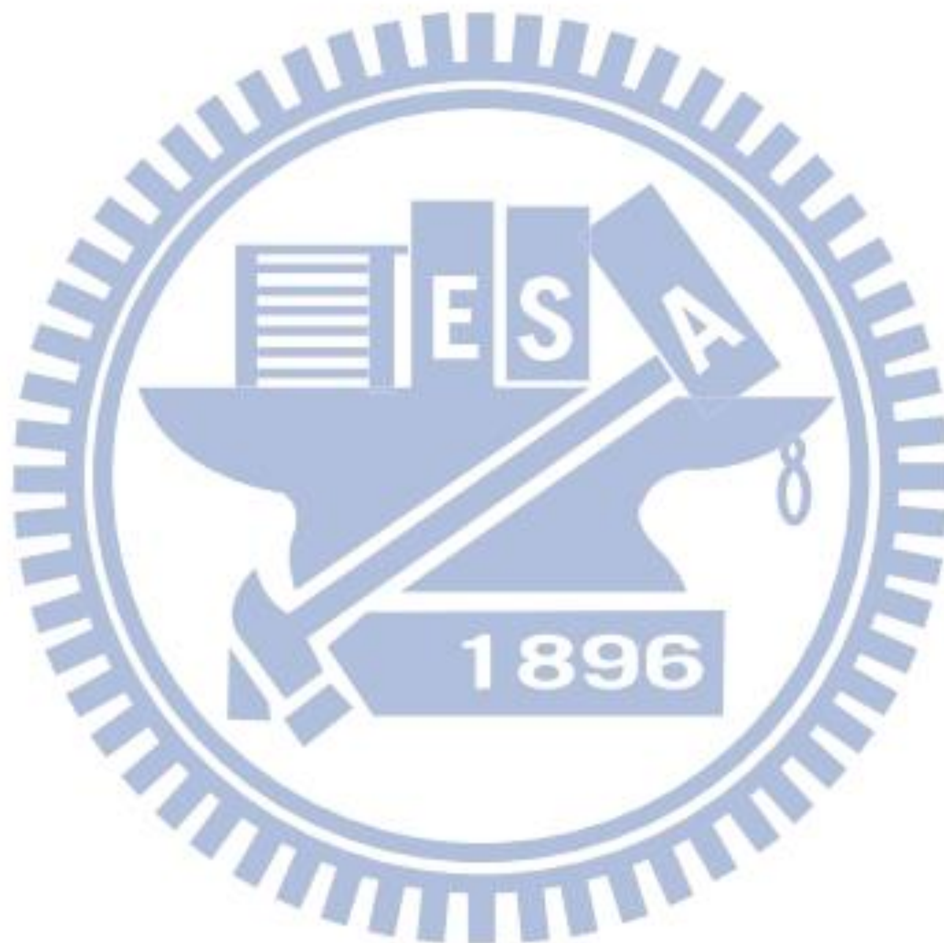
List of Tables

Table 1 – performance counters for ALPHA 21264 9
Table 2 – thermal difference between simulation and counter-based running program (lf) 20
Table 3 - thermal difference between simulation and counter-based running program (lffs) 21
Table 4 - functional units map to performance counters 27



List of Equation

Equation 1 - counter-based power model on real platform........................26



I. Introduction

Advanced CMOS process and three-dimensional structure technologies have been applied to IC design and production in recent years so that transistor density of microprocessors is increasing rapidly. Therefore, power consumption is one of primary design constraint for systems ranging from server computers to handhelds [1]. The power consumption of very large scale integration (VLSI) circuit can be categorized into dynamic power and static power which is also defined as leakage power. Then, the increasing of power consumption will lead to high operating temperatures and gradients of temperature between different locations of a microprocessor. Previous studies [24][25] indicated that the high operating temperature and temporal and spatial gradients of temperature on microprocessors increase the leakage power. Besides, it also results in clock skew, system failure, and reduce performance and reliability of microprocessors. As manufacturing technology improving, leakage power consumption increases more rapidly than dynamic power consumption and becomes a major concern in VLSI circuit design since the reducing in threshold voltage and increasing in operating temperature which is result by increasing in dynamic power consumption. Moreover, temperature is exponentially related to leakage power, and a positive feedback loop exists between temperature and leakage power (Figure 1), which can cause thermal runaway and damage the circuit [2]. Temperature control becomes a clear long-term threat to design technology in the next decade [3].

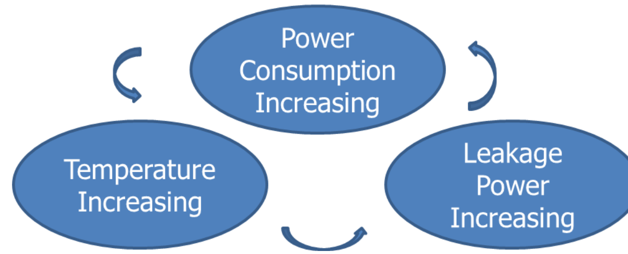


Figure 1 - feedback loop exists between temperature and power

As power and thermal problems in modern microprocessors become severe, a number of dynamic thermal management (DTM) technologies are proposed. These technologies can be divided into two categories, which are hardware-based and software-based management. Hardware-based management, such as clock gating, dynamic voltage and frequency scaling (DVFS) [12], requires additional hardware device equipped in the original system design and initialize the management behavior while certain power or thermal condition is satisfied. On the other hand, software-based management, such as task migration [7], task scheduling [23] and compiler optimization [8], can be applied in general systems without the support of management devices. However, software-based management may require additional data structures to maintain power or thermal states which would be applied for the management strategy. We will use these methods of management to adjust system operations to control the thermal behavior in real-time. However, before we design an efficient method of management, these technologies all require fast and accurate thermal information as inputs for determining the scheduling strategies. Imprecise or outdated temperature information may delay the activation of DTM and influence on the system operations.

In this thesis, related works are presented in chapter 2. In chapter 3, we will construct the whole simulation platform for microprocessors by using non-thermal sensor approaches to estimate the thermal information of the chip.

Chapter 4 shows the main idea of the method for speeding the flow of thermal estimation. In chapter 5, we would construct the thermal simulation on real platform to compare the result with the real thermal information through collecting temperature of the chip from the thermal measurement. Chapter 6 and 7 are the experimental setup and its results. The conclusions and future works are in chapter 8.

II. Related Works

In this research, we mainly focused on the methods of thermal estimation about how to obtain efficient and accurate thermal information for each functional unit. Thus, in following, we will describe related thermal estimation techniques in more detail.

The thermal estimation technologies could be divided into two major categories (Figure 2), which are hardware-based methods [14][15][16][17][18] and software-based approaches [19][20][21][22]. For hardware-based methods, which is also called thermal sensor approaches, thermal information is mainly obtained from on-chip thermal sensors. However, the hardware-based methods can only provide accurate results depend on there are sufficient numbers of accurate thermal sensors on a chip. Therefore, the hardware-based approaches may suffer from intrinsic accuracy problem of thermal sensors, manufacturing cost and die size if more sensors are allocated and location constraints of sensors on the chip [13]. On the other hand, power variation and thermal information can also be derived from the power models and thermal models over time through software-based approaches, which is also called non-thermal sensor approaches. For example, the power consumption of each component of

a chip could be obtained via computation from performance counters with power models, such as Wattch [11], McPAT [4], or direct measurement. Fine-grained thermal information could be estimated via using thermal models, such as Hotspot [10], to compute through applying the power consumption of each component of a chip as input. Wei et. al. [10] concluded that software-simulated thermal information can achieve very high accuracy. Nevertheless, simulation based on full thermal models would take significant execution time and memory usage which may not be suitable for DTM. There is an efficient thermal model, such as CLOFT 錯誤! 找不到參照來源。, to speed up thermal estimation and reduce memory consumption for the simulation process. In following, we will describe these two software-based thermal simulation techniques in detail.

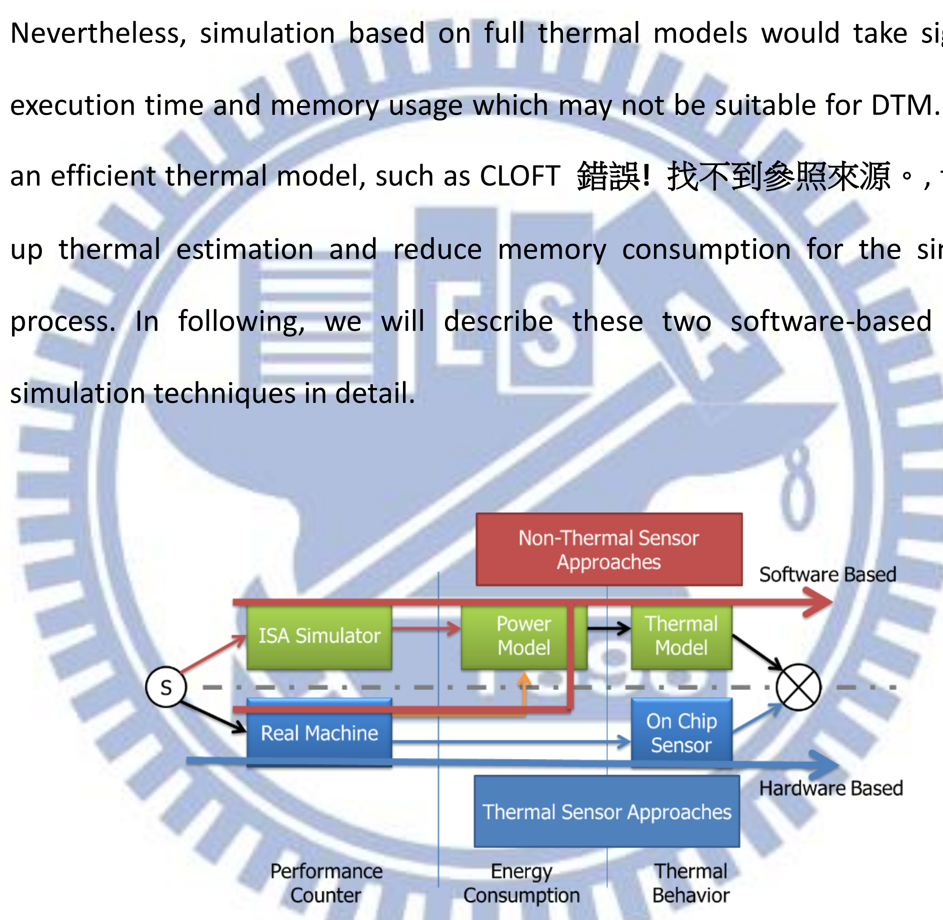


Figure 2 - classification of thermal estimation platform

About how to analyze thermal behavior of a microprocessor among these two methods of Hotspot and CLOFT for thermal model, both of they adopt a well-known way to map the phenomenon of heat transfer into electrical circuits [26]. Heat flow between two locations with a thermal resistance due to temperature difference is equivalent to current flow through an electrical

resistance because of a potential difference. Moreover, natural material contains certain heat capacity, such characteristic would be mapped into an electrical capacitance. The electrical circuits transform from the equivalent thermal behavior are called dynamic compact models (RC models). Adopting RC models is convenient for describing thermal behavior and deriving related phenomenon

Thermal analysis methods applied to integrated circuits can be divided into chip-level analysis [9][26][27][28][29] and architecture-level analysis [23][30][31][32][33][34]. Several modeling and simulation tools are also developed in architecture-level. Hotspot and CLOFT are architecture-level thermal simulators because its accuracy was validated and its efficiency is better than chip-level analysis [10]. However, the main difference of Hotspot and CLOFT is their complexity of constructed dynamic compact model. Hotspot adopts fourth order Runge-Kutta method [35] and derives the fourth order differential equations for constructed dynamic compact, which leads to high overhead in computation and memory usage, to solve these differential equations. Compare to the method of CLOFT, its electrical circuit model is reduced to estimate the thermal behavior of a microprocessor which significantly reduces the computation and memory space with a slight degradation of the accuracy. They just have 0.3~1.5% thermal difference, but CLOFT get about 34~47times speed up and 0.45% memory usage.

III. Simulation Platform for Microprocessors

In this chapter, we introduce how to build a simulation platform to estimate thermal in general and the architecture of platform. After that, we

analyze performance, power consumption, and thermal estimation under different technology size by using different compiler optimization through applying our simulation platform as evaluated tool. We will get something different results compared to the traditional view about the compiler design, and It is beneficial to adjust the design way to enhance energy saving.

3.1 Architecture of Simulation Platform

We will run the workloads on ISA simulator and modify it to get corresponding each functional unit performance counters of the chip. Then, if we get the performance counters as understanding the program behavior, we will use power model to calculate the power information of each functional units through applying performance counters as input. Next, each functional unit thermal information of the chip will be obtained from using thermal model to calculate through applying power information as input.

Figure 3 shows our detailed flow chart of simulation platform. We use Gem5 for our ISA simulator to get performance counter and McPAT for our power model to get power information. Between Gem5 and McPAT we design a parser program to let the output of performance counters from Gem5 to be read by McPAT. At last, we apply Hotspot as our thermal model to get thermal information through using power information from McPAT to calculate. We will introduce Gem5, McPAT, and Hotspot in detail as follow.

Gem5 [9] is a well-known ISA for computer architecture proposed in 2011. A pure workload behavior without OS scheduler can be obtained via modifying gem5 system to provide performance data in each sample period. Gem5 can build several binaries for different guest architecture, simulation mode, and use. Currently, the available architectures are ALPHA, ARM, MIPS, POWER, SPARC,

and x86. In our thesis, we adopt ALPHA architecture in the experiments.

McPAT [4] is an integrated power, area, and timing modeling for multicore, and many core architectures, which is used to compute dynamic power and leakage power consumption and support comprehensive early stage design space exploration for multicore, and many core processor configurations ranging from 90 nm to 22 nm. In the experiment, McPAT obtains performance counters of each functional block from gem5 and produce dynamic power of each functional block in each sample period. Each dynamic power value is adopted as input for thermal simulator.

Hotspot [10], an architecture level thermal simulator built by Virginia University, is designed to compute temperature profiles of different setting of microprocessors. It can be used in conjunction with power simulators, such as Wattch or McPAT. Hotspot is compatible with different kinds of power and performance models without requiring detailed design or synthesis description.

In our simulation platform for microprocessor, we can compute power information of each functional unit we want in cycle accurate. Moreover, we will use output of temperature from Hotspot to calculate leakage power and let the leakage power as the next cycle input power to calculate the new temperature because we will consider the influence of leakage power on temperature. At last, along with the development of the integrated circuit(IC) technology, we will analyze the each generation technology size from 90 nm~22 nm, not just for the default technology size 130 nm.

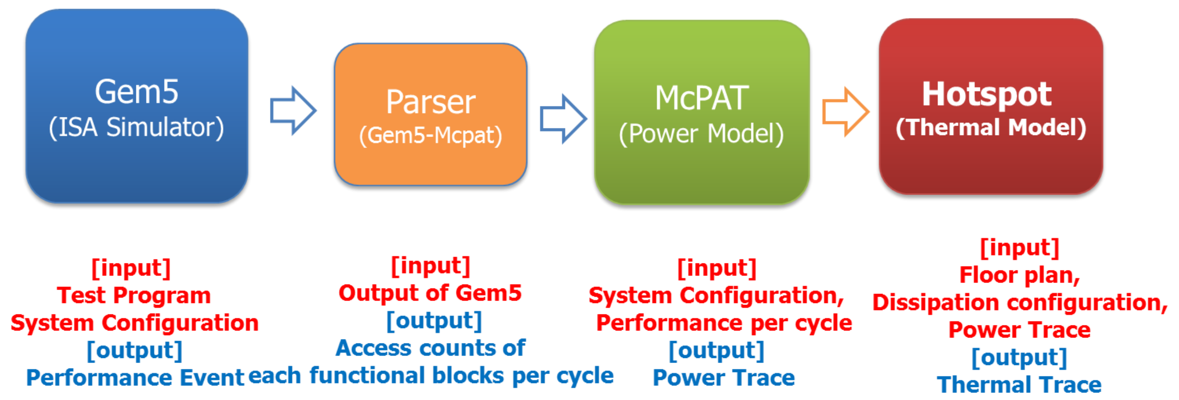


Figure 3 – flow of simulation platform

3.2 Thermal Estimation under Different Technology Size Using Different Compiler Optimization

In this part, we will analyze the results about running the same workload under different technology size through using different compiler optimization. First, Instead of using customized program, we use 401-Bzip2 of SPECCPU2k6 benchmarks as our workloads and ALPHA 21264 is adopted for our microprocessor architecture when we use Gem5 ISA simulator with CPU frequency setup 2GHz and generate much meaningful power information with sample rate 10000 cycles. The floor plan of ALPHA 21264 is shown in Figure 4. Because we want to get power and thermal information of theses functional units, Table 1 is shown the type of performance counters we catch to use. For the different technology size of the chip we will use 70 nm to compare with 22 nm. Under these two different size (70 nm and 22 nm) of technology we respectively apply O1 and O2, which are different types of compiler

optimization, to compile Bzip2 to as their workloads and analyze their performance, power information and thermal information. O1 is one type of compiler optimization to minimize the code size and O2 is another one type to maximize speed.

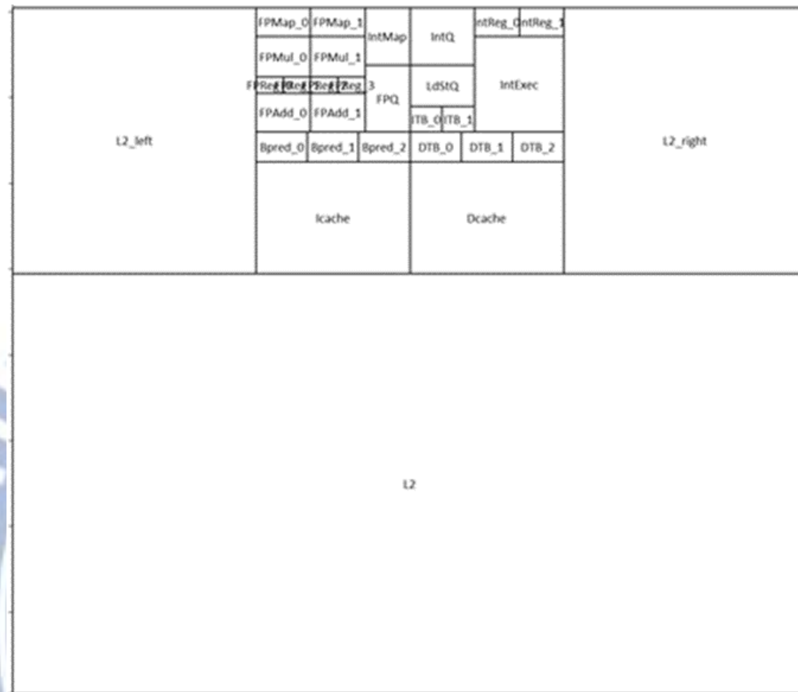


Figure 4 - ALPHA 21264 processor floor plan

Table 1 – performance counters for ALPHA 21264

Performance Counters for ALPHA21264
BRANCH_MISS
BRANCH_INSTR
BRANCH_RASUSE
LSQ_PREG
LSQ_WAKEUP
INT_REG
FP_REG

INT_ALU
FP_ALU
ICACHE_miss
DCACHE_miss
ICACHE_hit
DCACHE_hit
L2CACHE_miss
L2CACHE_hit

Thermal information about integer/floating register, icache/dcache, and integer/floating ALU under 70 nm using O1 and O2 is shown in Figure 5 and another situation of thermal information under 22 nm using O1 and O2 is shown in Figure 6. We can see the different detailed thermal information of each functional unit between different technology sizes. Comparing the result of O1 with O2 under these two technology sizes, we can find that the average temperature of O2 is higher than the result of O1 under both of 70 nm and 22 nm because O2 will maximize speed and it will let the core not only enhance its performance but also its temperature.

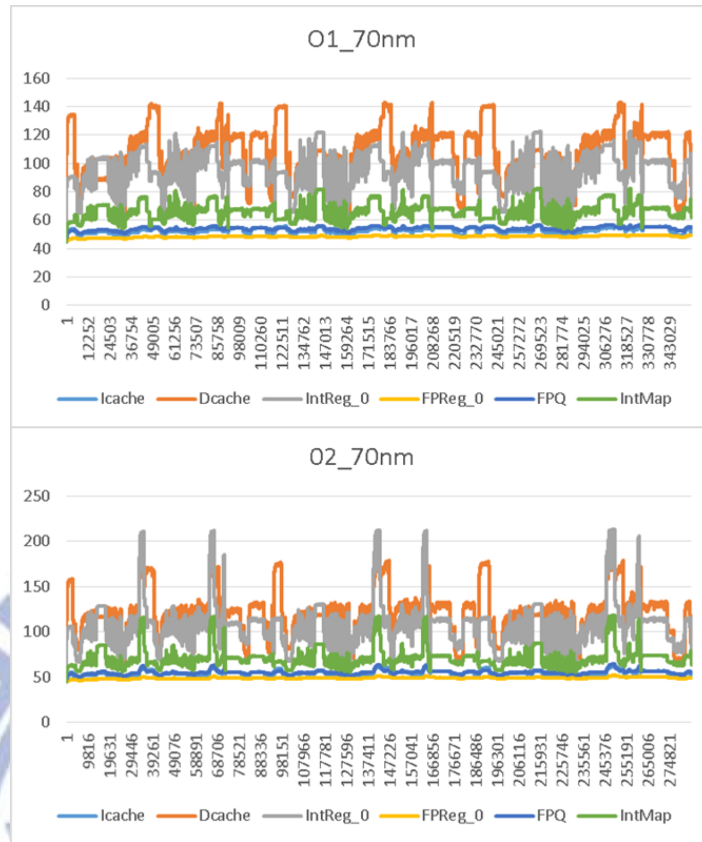


Figure 5 – thermal information of functional units when run Bzip2 using O1 and O2 under 70 nm

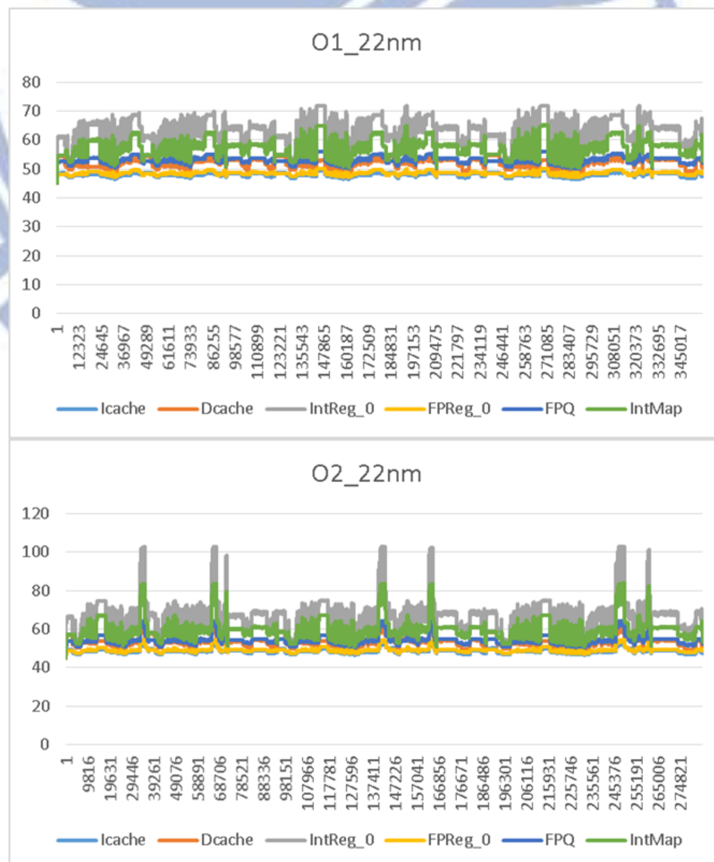


Figure 6 - thermal information of functional units when run Bzip2 using O1 and O2 under 22 nm

Through observing Figure 7 we can know that O2 has less execution time than O1 and it means O2 has better performance. In Figure 8 and Figure 9 it is shown dynamic power and dynamic energy under 70 nm and 22 nm through using O1 and O2 respectively. Because O2 has better performance, we can get O2 consume much more dynamic power than O1. But if we consider the execution time between O1 and O2 to compute their dynamic energy, we find that O1 accumulate much more dynamic energy consumption than the result of O2 since O1 have to perform more long. Hence, we can get that in traditional way if we optimize the performance, we also can get less power energy consumption so we only need to optimize its speed.

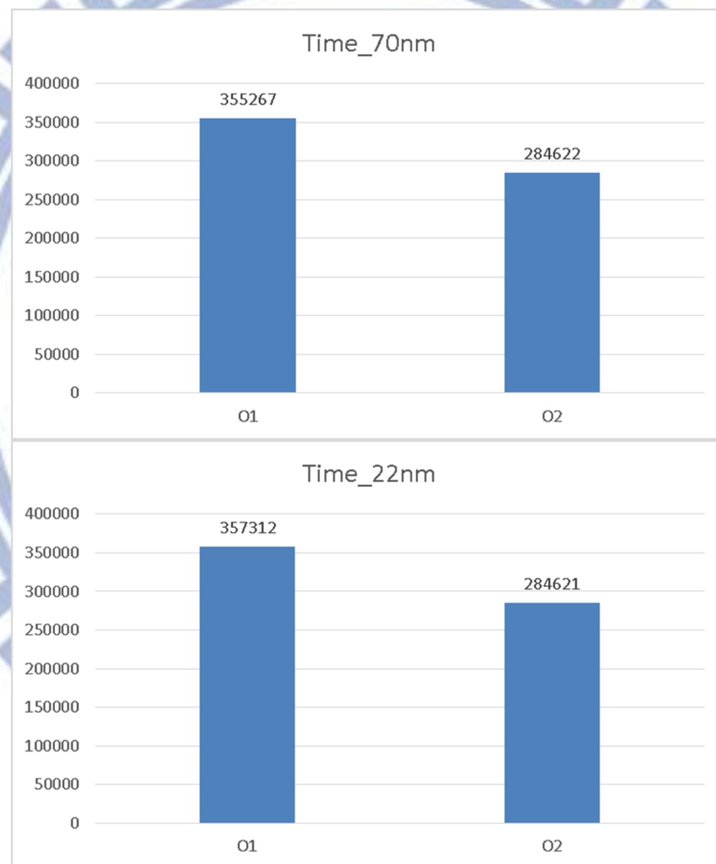


Figure 7 – Execution time under 70 nm and 22 nm using O1 and O2 respectively

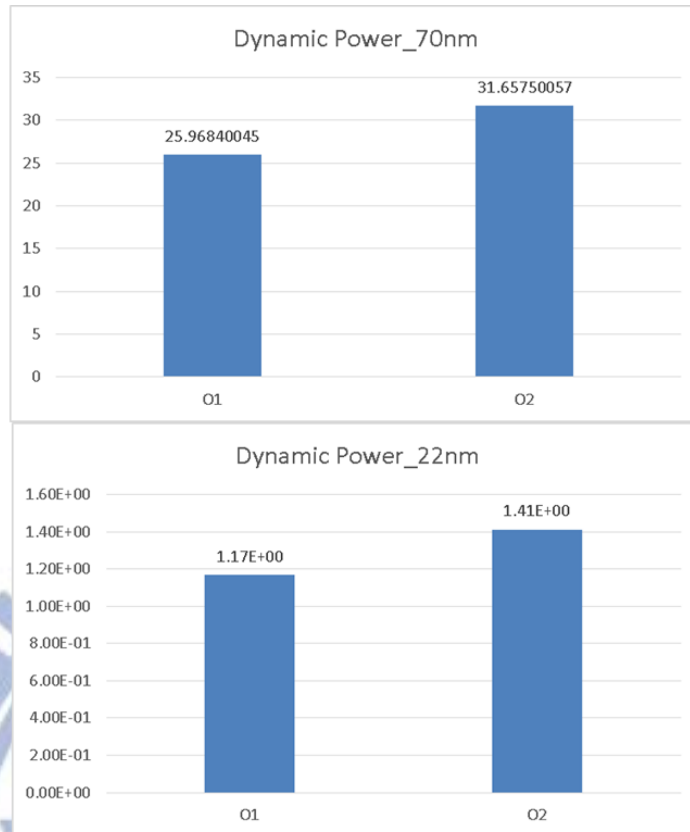


Figure 8 – dynamic power under 70 nm and 22 nm using O1 and O2 respectively

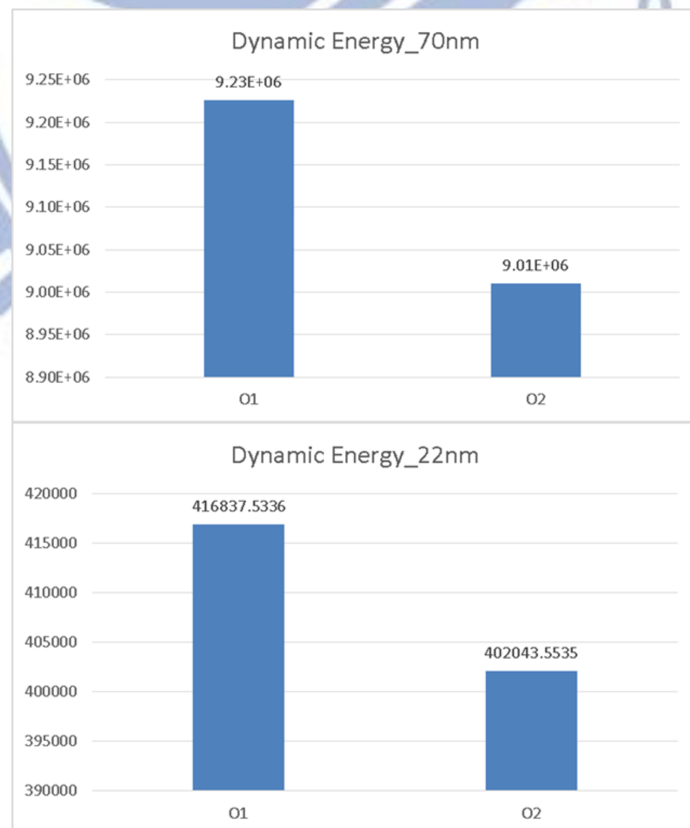


Figure 9 - dynamic energy under 70 nm and 22 nm using O1 and O2 respectively

We know that dynamic power almost occupy the whole power consumption in traditional way, so we just need to notice dynamic power consumption. However, if we consider the new generation technology size under 90 nm, we can find that leakage power will occupy a big part of power consumption in the future technology. Figure 10 is shown that the proportion of dynamic power and leakage power under 70 nm and 22 nm using O1 and O2 respectively. Along with the decreasing of technology size, the proportion of leakage power in total power consumption has risen from 50% to 90% among 70 nm and 22 nm. Leakage power consumption will accounts for up to 90 percent of the total power consumption in the future.

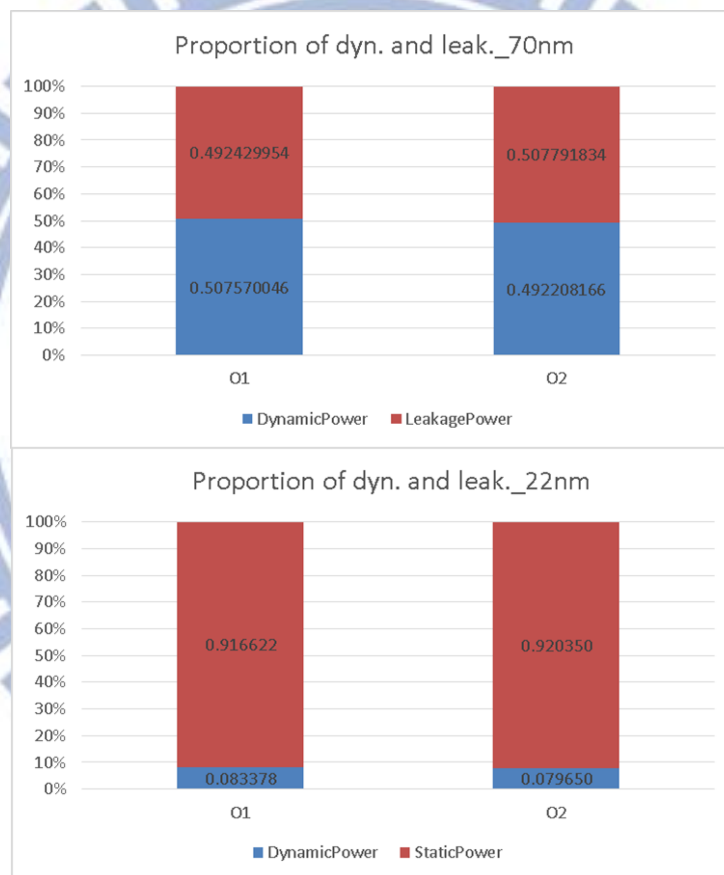


Figure 10 – proportion of dynamic power and leakage power under 70 nm and 22 nm using O1 and O2 respectively

Hence, we observe leakage power and energy consumption under 70 nm and 22 nm. Figure 11 and Figure 12 show that O2 not only consume much more leakage power but also leakage energy than the result of O1. Because leakage power consumption will accounts for up to 50 percent of the total power consumption under 90 nm and the difference of leakage power between O1 and

O2 will be much bigger along with the decreasing of technology size, the influence of performance can't let the leakage energy consumption of O2 to be smaller than the result of O1.

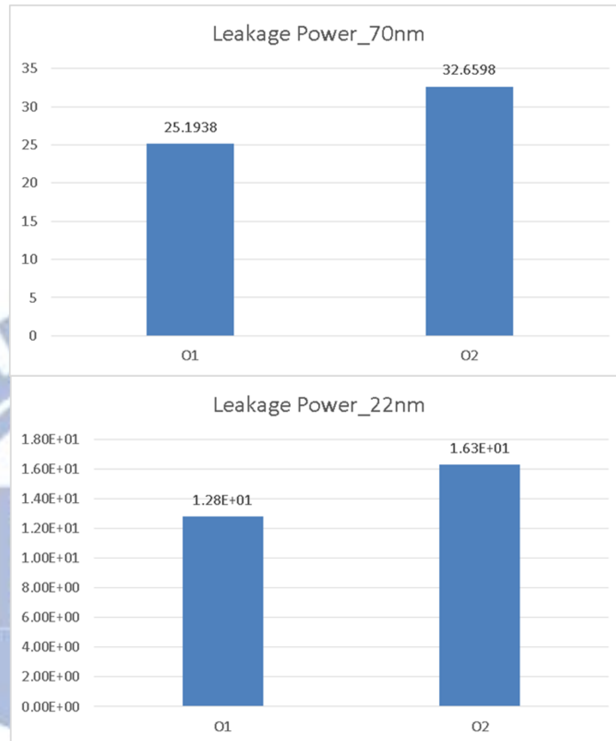


Figure 11 – leakage power under 70 nm and 22 nm using O1 and O2 respectively

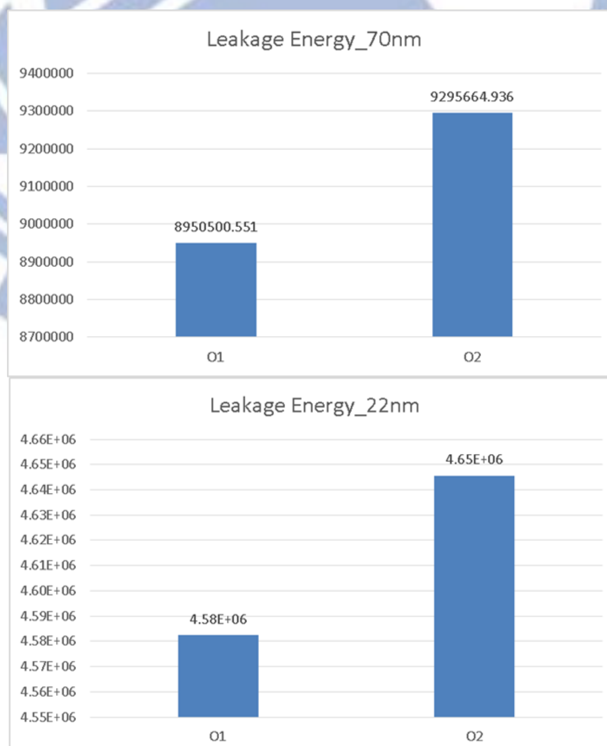


Figure 12 - leakage energy under 70 nm and 22 nm using O1 and O2 respectively

Figure 13 and Figure 14 show the results of total power and energy consumption under 70 nm and 22 nm using O1 and O2 respectively. Because of the influence of leakage power and energy, it results in that the total power and energy of O2 is much bigger than the result of O1. In the future, we can't just ignore the influence of energy consumption since it is not right that if we optimize performance, then it will optimize the energy consumption by increasing speed.

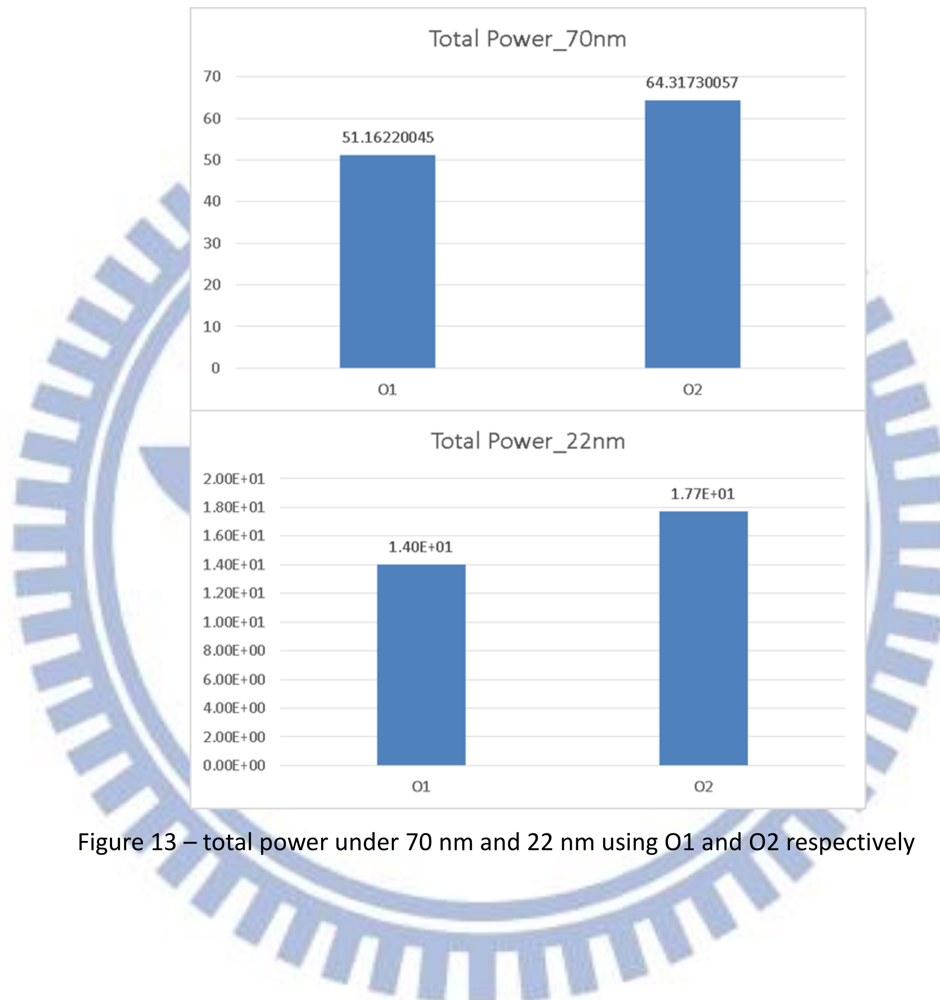


Figure 13 – total power under 70 nm and 22 nm using O1 and O2 respectively

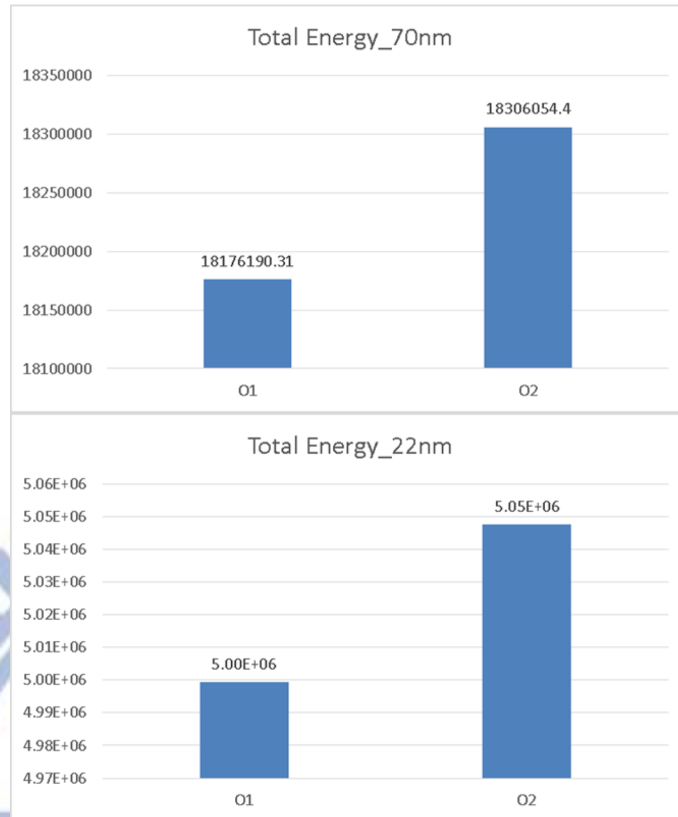


Figure 14 - total energy under 70 nm and 22 nm using O1 and O2 respectively

At last, we will analyze the energy-delay product (EDP) [6] of O1 and O2 under 70 nm and 22 nm. The results are shown in Figure 15. We can see that although the EDP of O2 is better than the result of O1 no matter we use 70 nm or 22 nm, we can see the results in Figure 16. It is shown that the difference of EDP between O1 and O2 is much smaller along with the decreasing of technology size. If the technology size of IC keep reducing in the future, the difference of leakage power and energy consumption between O1 and O2 will be more enormous and the EDP of O1 will be better than the EDP of O2 since the influence of leakage energy on EDP will be much heavier than the delay. Hence, we can’t just see how to optimize the performance but we have to focus on the influence of leakage power and energy consumption in the next decade compiler design.

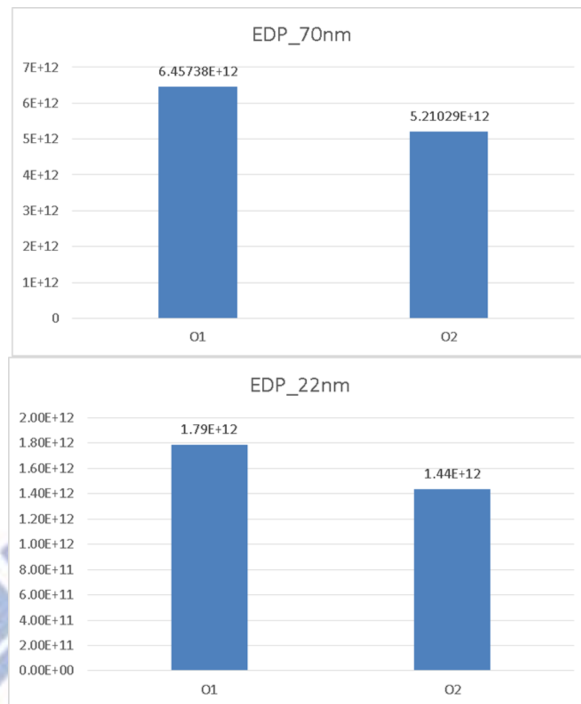


Figure 15 – energy delay product under 70 nm and 22 nm using O1 and O2 respectively

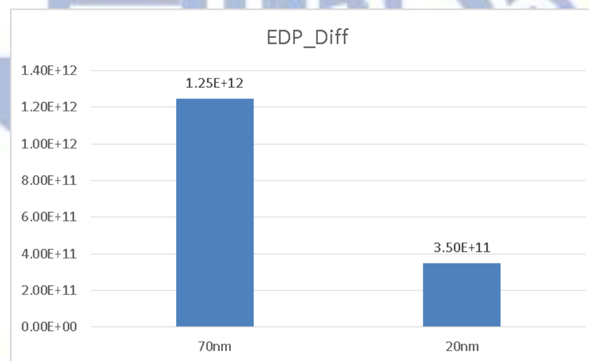


Figure 16 – the difference of EDP between O1 and O2 under 70 nm and 22 nm

IV. Counter-Based Approach

After we successfully construct the simulation platform for microprocessors to do thermal estimation, we find that there is a problem of simulation. For our simulation platform, if we want to know the thermal distribution between each functional unit when we run a workload, we have to use gem5 to execute the workload to get performance counters and let power model and thermal model to calculate power and thermal information. There is a big problem is the speed of simulation platform. If we want to do thermal estimation when we execute Bzip2 of SPECCPU2K6 benchmarks, it have to cost

3 hours to get the last thermal information of each functional units by using the whole simulation platform. But actually the execution time of Bzip2 for normal processor is just need to cost 3 minutes. The execution time of simulation platform to finish thermal estimation is much slower than the actual execution time of the workloads. Hence, this problem will result in that we can't get thermal information in real time.

In this chapter, we will propose a counter-based approach as our power model to calculate power information so that it can speed up the simulation platform. In Figure 17, it is divided into two ways, simulation-based and counter-based platform. For simulation-based platform, it is our original simulation platform and we will use McPAT and Hotspot as power model and thermal model. For counter-based platform, we will use our counter-based approach as power model. It is a way to use regression modeling strategies for performance counters of each functional units and current power information of chip. With regard to thermal model, we will use method of CLOFT to speed up the calculation of thermal information. Next, we will verify the accuracy of counter-based platform with simulation-based platform.

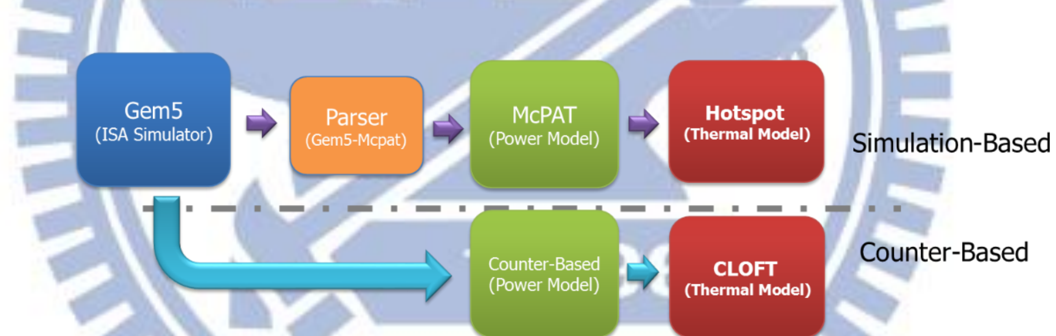


Figure 17 – simulation-based and counter-based platform

4.1 Verification

First, we will train the counter-based power model by using regression modeling strategies. Then, we will use two micro benchmark programs as the workloads to compare the results of counter-based platform with the results of simulation-based platform. The first micro benchmark program (If) is designed to interleaving execute the integer and floating point operations. Figure 18 shows the result of thermal information of each functional unit when we use (If) as the workload between simulation-based and counter-based platform. The rate of thermal difference is shown in Table 2under different functional units and its average rate of thermal difference is 0.822%.

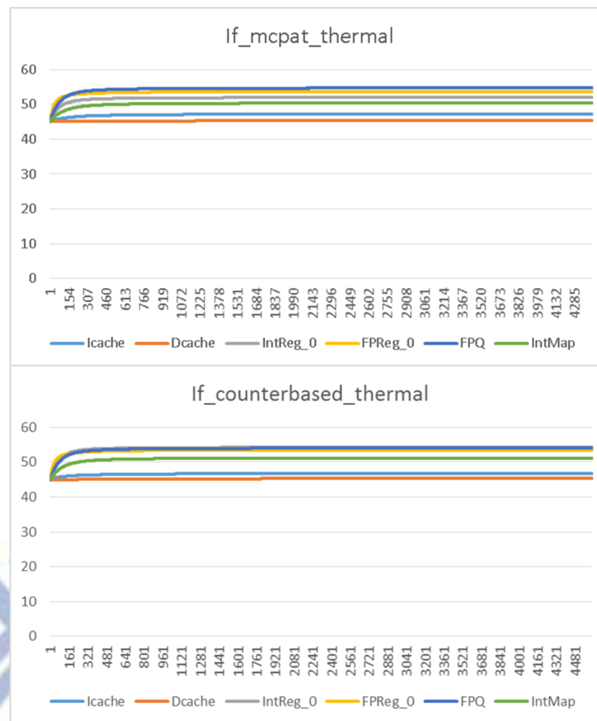


Figure 18 – thermal information of functional units when run program If

Table 2 – thermal difference between simulation and counter-based running program (If)

Micro benchmark program (If)	
Functional Units	Rate of thermal difference
Icache	0.873%
Dcache	0.055%
Integer register	0.874%
Floating register	0.351%
Floating ALU	1.152%
Integer ALU	1.629%

And the second micro benchmark program (Iffs) is designed to keep executing the integer operations and then keep executing the floating point operations. The result of thermal information of each functional unit when we use (Iffs) as the workload between simulation-based and counter-based platform is shown in Figure 19. The rate of thermal difference is shown in Table 3 under different functional units and its average rate of thermal difference is 2.52%.

Therefore, under the smaller thermal difference between simulation-based and counter-based platform, we can speed up the simulation platform by using counter-based approach to get similar thermal information.

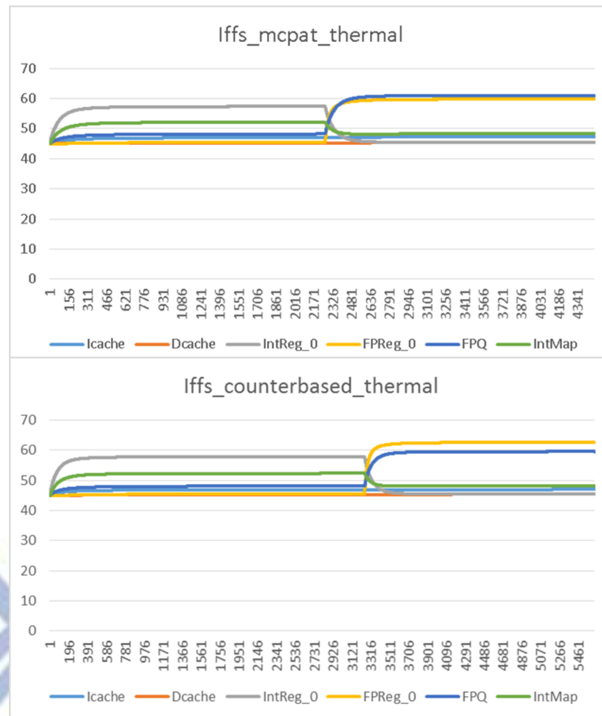


Figure 19 - thermal information of functional units when run program Iffs

Table 3 - thermal difference between simulation and counter-based running program (Iffs)

Micro benchmark program (Iffs)	
Functional Units	Rate of thermal difference
Icache	0.597%
Dcache	0.072%
Integer register	0.597%
Floating register	6.183%
Floating ALU	5.584%
Integer ALU	2.085%

V. Thermal Simulation and Measurement on Real Platform

In this chapter, we will doubt the correctness of thermal simulation platform for microprocessor so we will verify its correctness by comparing the result of thermal information from simulation platform with the real thermal information on physical system. If we want to get the real thermal information, we have to directly measure the real chip through taking its infrared image to

get different temperature data among each functional unit. Hence, we have to construct thermal simulation and measurement on real platform. Next, we will introduce our architecture of thermal simulation on real platform and how to do thermal measurement on real platform.

5.1 Thermal Simulation on Real Platform

In Figure 20, this is the flow chart of thermal simulation on real platform. We can see orange line in figure and get that we use real machine instead of ISA simulator to obtain the performance counters of each functional unit. Then, we equally apply power model and thermal to calculate power information and thermal information.

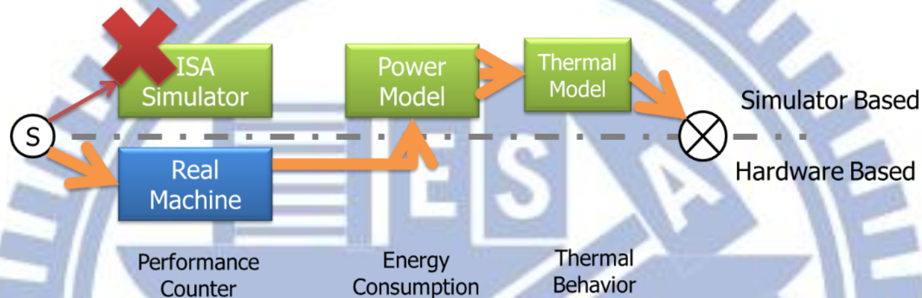


Figure 20 – flow of thermal simulation on real platform

So, the architecture of thermal simulation on real platform is shown in Figure 21. We will use *A* core *NXXXX* as our real machine and then run a test program on it to obtain the performance counters of each functional unit. Immediately after getting the performance counters, we will use it as input to calculate power information through applying counter-based approach as power model. At last, we can apply the method of Hotspot as thermal model to obtain the thermal information of each functional unit.

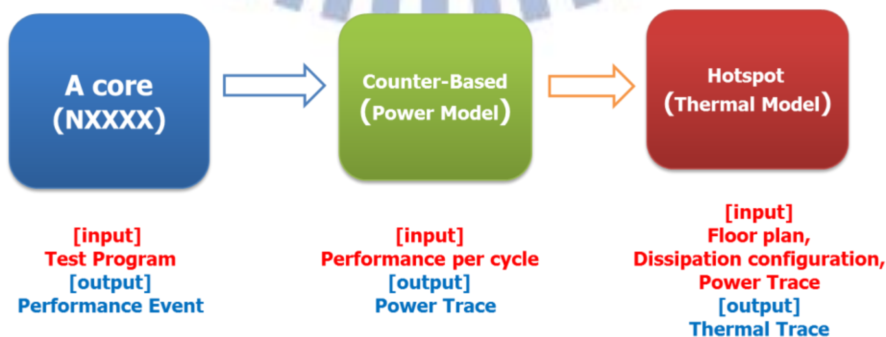


Figure 21 – architecture of thermal simulation on real platform

5.2 Thermal Measurement on Real Platform

In this part of getting real thermal information on physical system, the flow chart of thermal measurement on real platform is shown in Figure 22. First, we can't use ISA simulator anymore and apply real machine instead of it because we have to directly obtain real thermal information on physical system. We will also use A core NXXX as our real machine then it can compare the result with simulation way. Next, if we want to get real thermal information of each function unit on chip in detail, we will use infrared camera to get the whole thermal map of the chip instead of using on-chip sensor to get partial thermal information. Hence, we can see purple line in figure and get that we can take infrared image of the chip to get real thermal information of each functional unit in detail by using infrared camera to implement it.

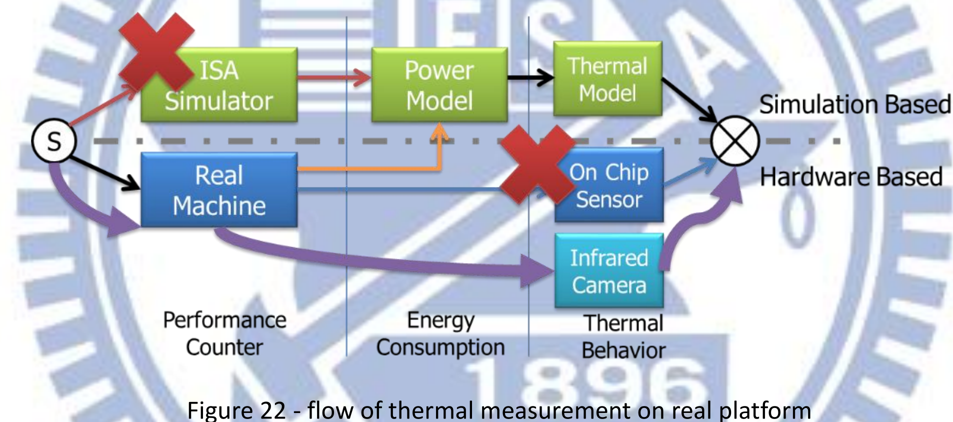


Figure 22 - flow of thermal measurement on real platform

VI. Experimental Setup

The experimental environment and experiment setup are introduced in this chapter. We will show floor plan of *A* core *NXXXX*, detail information of infrared camera, and how to construct counter-based power model on real platform.

6.1 Floor Plan of Core

In Figure 23, we can get the floor plan of the chip in left part of figure and we transform it into the simplified version in right part of figure. The floor plan of the whole chip is divided into four parts: Sub-System A, Sub-System B, Blank,

and GPU. Both of Sub-System A and B are cores. It is a dual core chip and make from 90 nm of technology size. The floor plan of core is shown in Figure 24. The left part of figure is the simple floor plan of the core, and we transform and divide it into six parts of functional units, which is shown in right part of figure. These six functional units are Load-Store Unit (LSU), Arithmetic Logic Unit (ALU), Data Cache (DCACHE), Memory Management Unit (MMU), Branch Target Buffer (BTB), and Instruction Cache (ICACHE).

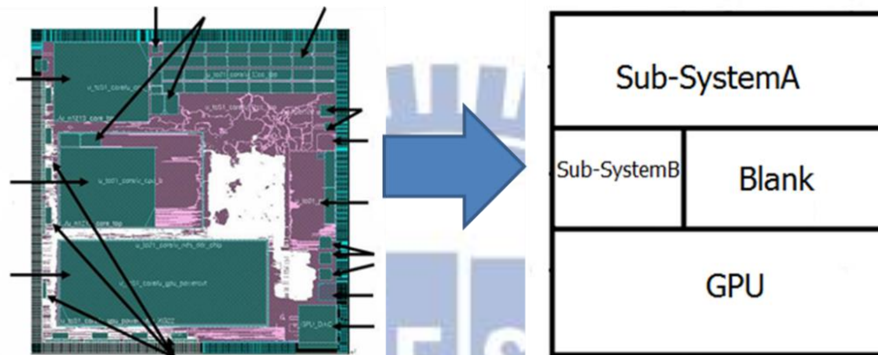


Figure 23 – floor plan of the chip

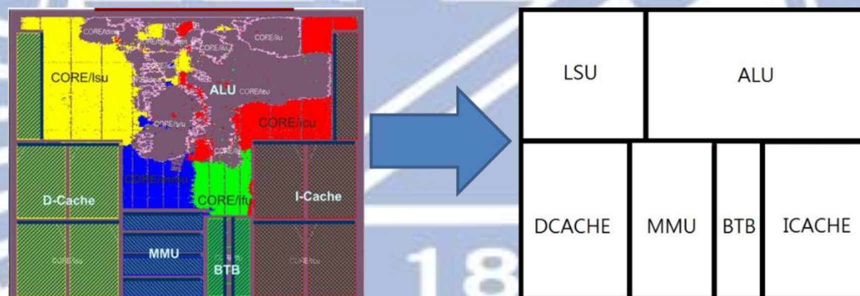


Figure 24 – floor plan of the core

So, we combine the floor plan of core into the floor plan of chip. The entire floor plan is shown in Figure 25. We will observe and compare the thermal information of these six different functional units on the core with the real thermal information which is obtained from infrared camera.

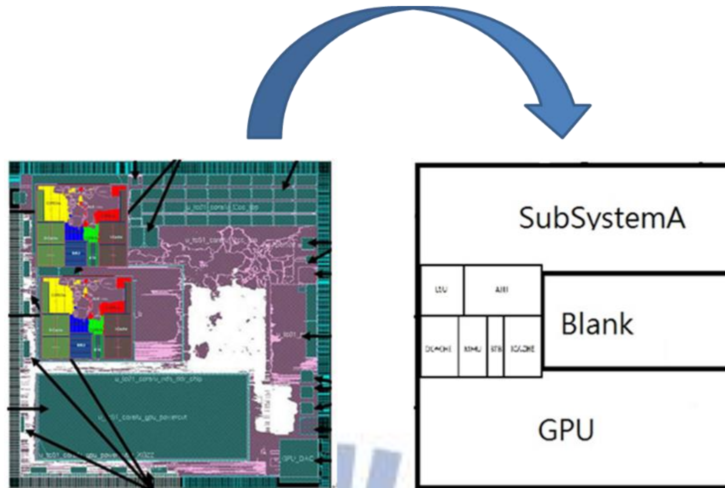


Figure 25 – entire floor plan of chip

6.2 Infrared Camera

For infrared camera, we use FLIR SC7000 which is shown in Figure 26. It can produce crisp thermal images of 640 * 512 pixels and 20 mK thermal sensitivity captures the finest image details and temperature difference information. Depending on the model and detector, the FLIR SC7000 can deliver thermal images up to a speed of 62000 Hz. Windowing allows a subset of the total image to be selectively read out with user-adjustable window size at a much higher frame rate. The sub-sample window sizes and locations can be randomly chosen and are easily defined using official camera control software. Figure 27 shows that our environment setup to use infrared camera to capture the thermal map of the chip and the bottom left corner of the figure shows the view it capture.

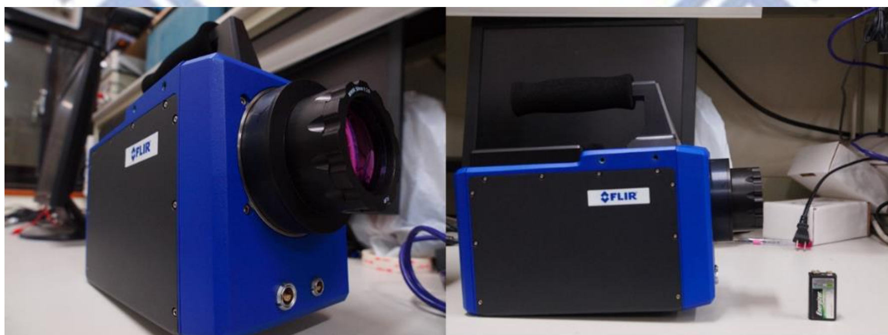


Figure 26 – FLIR SC7000

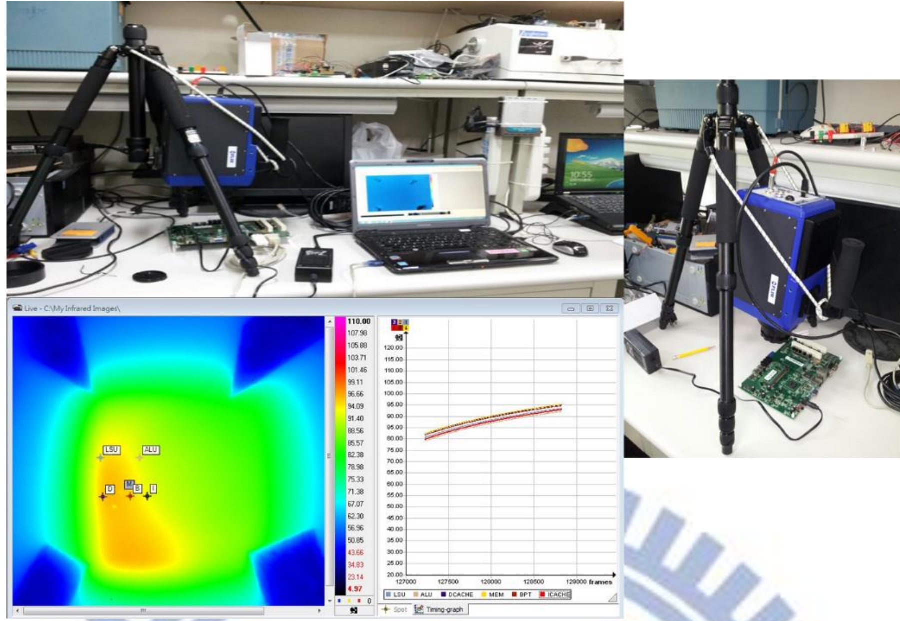


Figure 27 – environment setup for FLIR SC7000

6.3 Counter-Based power model on real platform

In this part, we will construct counter-based power model for *A* core *NXXXX* by using the regression modeling strategies for performance counters of each functional unit and current power of the chip. Equation 1 is the equation for regression. $\mathrm{P_{current}}$ is current power of the chip. $Perf_x$ is the performance counters where x is depend on the number of each performance counter you choose. α_y is the answer we want to solve and it is parameter multiplied by each performance counter where y is depend on different number of performance counters. In Table 4, we can get these performance counters of chip mapping to each functional unit. Then, we will capture current power of the chip by using leak current fluctuations monitor in Figure 28. Then, we can use the result of regression to calculate power information of each functional unit in real time.

$$\mathrm{P_{current}} = \mathrm{C} + \alpha_1 * Perf_1 + \alpha_2 * Perf_2 + \cdots where\ C\ is\ const.$$

Equation 1 - counter-based power model on real platform

Table 4 - functional units map to performance counters

Functional Units	Performance Counters
LSU	Loads completed/stores completed
ALU	Multiply instructions
DCACHE	Data cache access times
MMU	Hardware-assisted page table walker request
BTB	Taken conditional branches times
ICACHE	Instruction cache access times

Figure 28 - capture current power of chip

VII. Experimental Results

In this chapter, we present two parts of experimental results, thermal visualizer from result of thermal measurement, and thermal simulation. And to compare thermal information of each functional unit from thermal simulation with the measurement result from infrared camera.

In the first part, Figure 29 and Figure 30 are respectively show the thermal visualizer of result from measurement and simulation. The view of thermal

measurement is supplied by the official camera software so we write the thermal visualizer by ourselves to show the thermal information of thermal simulation which can be compared with the view of thermal measurement.

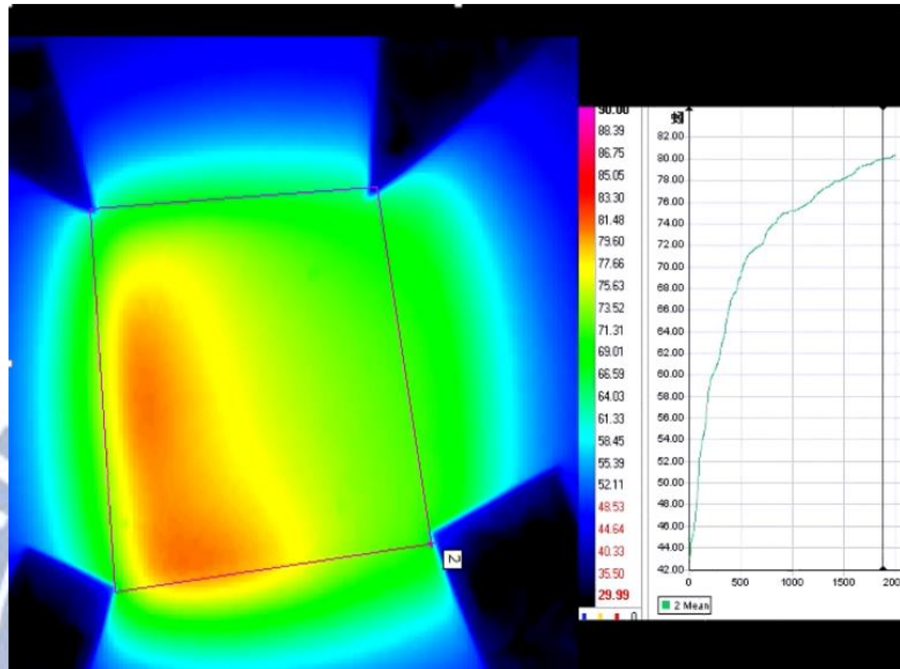


Figure 29 – thermal visualizer of thermal measurement

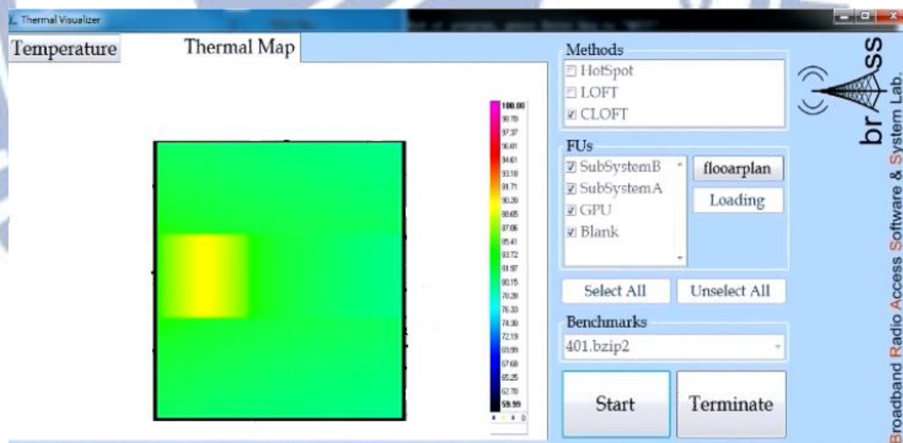


Figure 30 – thermal visualizer of thermal simulation

In the second part, we use SPECCPU2k6 benchmarks and choose 401.bzip2, 429.mcf, and 450.spolex as our workloads. Then, we run these workloads on the A core NXXXX respectively. Next, we compare the result of thermal simulation with thermal measurement under different workloads running on *A* core *NXXXX*. The comparative results are shown in Figure 31, Figure 32, and Figure 33. Although they don't have the equal thermal information of each functional unit, but they have the same trend of thermal information between each functional unit. We can see the red circle to know that they get the phase change at the

same time and show the similar trend of the change of thermal information in each functional unit.

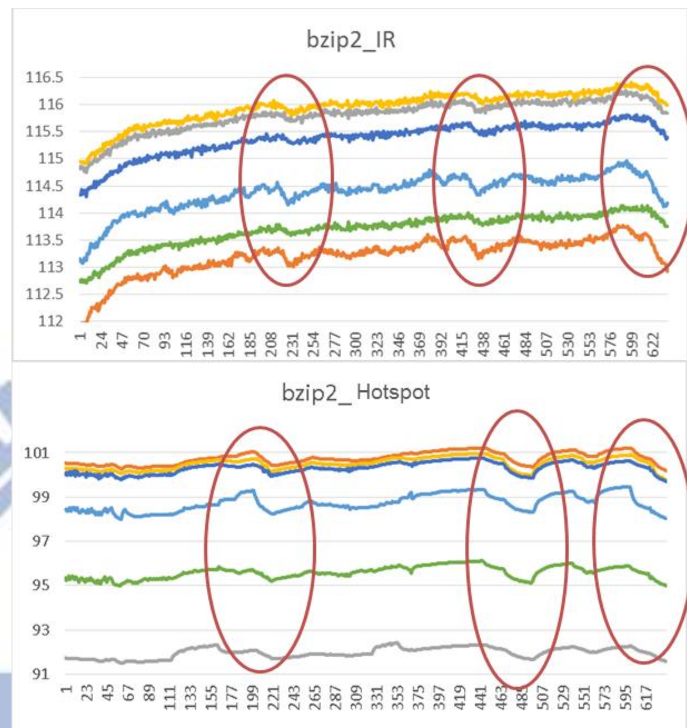


Figure 31 – thermal information of thermal measurement and simulation running bzip2

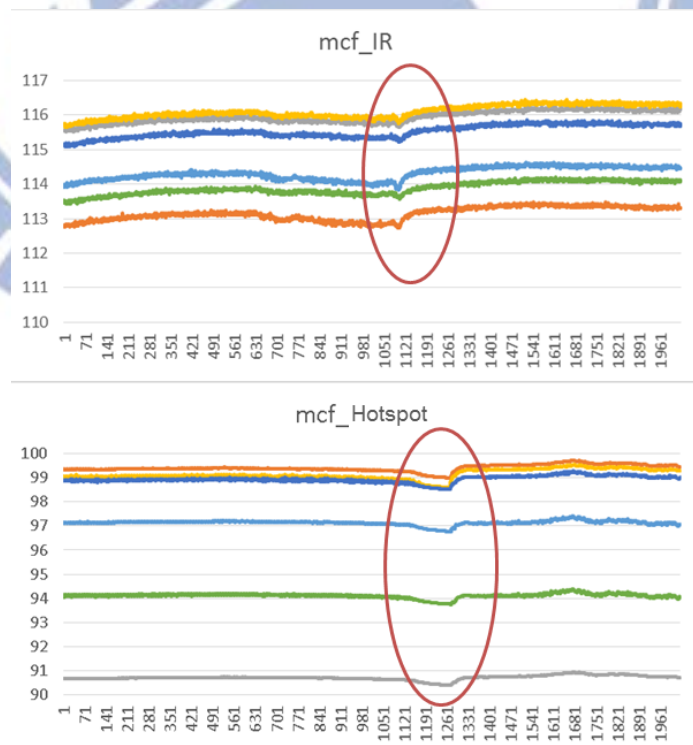


Figure 32 - thermal information of thermal measurement and simulation running mcf

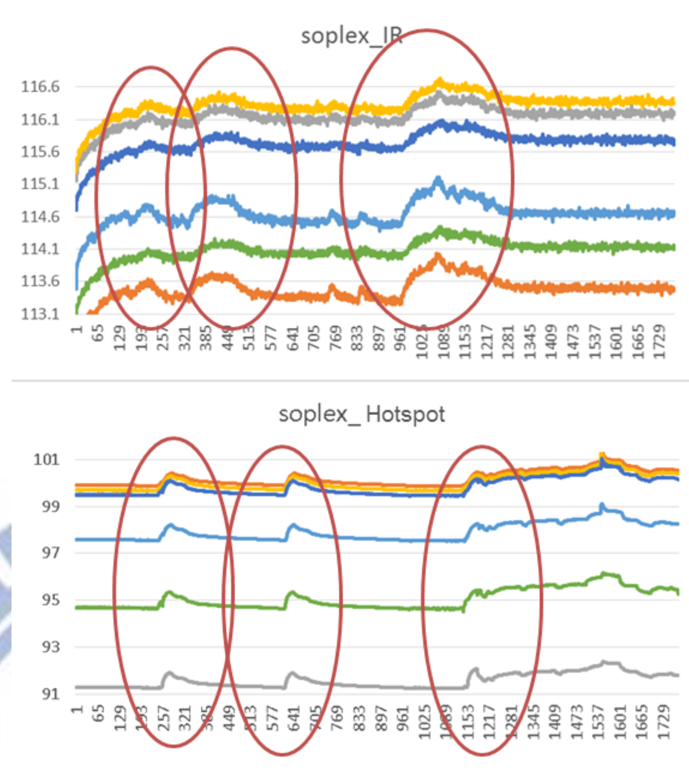


Figure 33 - thermal information of thermal measurement and simulation running soplex

VIII. Conclusions and Future Works

In this study, we construct thermal simulation platform for microprocessor. It can cycle accurate to compute power consumption, calculate leakage power consumption generated from different temperature, and be used in each generation technology size from 22 nm~130 nm. Then, we propose counter-based approach to compute power consumption and it can speed up the thermal estimation. At last, we do the verification between thermal simulation and thermal measurement on real platform. About the future works, we can explore the compiler design for next generation chip because of the influence of leakage power on total power consumption along with the decreasing of technology size. We can't just think the performance, but it need

to take energy consumption into consideration. Moreover, we can analyze the parameter for thermal simulation on real platform in detail. It will let the results of thermal simulation and measurement to be closer.

Reference

[1] S. Borkar, "Design challenges of technology scaling," IEEE Micro, pp. 23–29, Jul.-Aug. 1999.

[2] L. He et al., "Considering the Interdependence of Temperature and Leakage Interdependence of Temperature and Leakage," DAC, 2004.

[3] International Technology Roadmap for Semiconductors (ITRS-09). http://www.itrs.net/Links/2009ITRS/Home2009.htm

[4] S. Li, J. Ahn, J. B. Brockman, and N. P. Jouppi, "Mcpat 1.0: An integrated power, area, and timing modeling framework for multicore architectures," HP Labs, 2009.

[5] Pei-Shu Huang, Shiao-Li Tsao, Quan-Chung Chen, and Chen-Wei Huang, "An Efficient Thermal Estimation Scheme for Microprocessors," The 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2014).

[6] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low Power Digital Design," in IEEE International Symposium on Low Power Electronics, October 1994, pp. 8–11.

[7] H. Seongmoo, K. Barr, and K. Asanovic, "Reducing power density through activity migration," 2003 International Symposium on Low Power Electronics and Design, pp. 217-222, 2003.

[8] M. Mutyam, F. Li, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Compiler-directed thermal management for VLIW functional units," Conference on Languages, Compilers, and Tools for Embedded Systems(LCTES),, pp. 163-172, 2006

[9] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," SIGARCH Comput. Archit, News, pp. 1-7, August, 2011.

[10] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: acompact thermal modeling methodology for early-stage VLSI design," IEEE Trans. Very Large Scale Integr, Syst., pp. 501-513, May, 2006.

[11] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for

architectural-level power analysis and optimizations," In Proceedings of the 27th annual international symposium on Computer architecture (ISCA '00), ACM, New York, NY, USA, pp. 83-94, 2000.

[12] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," In Proceedings of the 33rd annual international symposium on Computer Architecture (ISCA '06), IEEE Computer Society, Washington, DC, USA, pp. 78-88, 2006.

[13] A. Bakker and J. Huijsing, "High-Accuracy CMOS Smart Temperature Sensors," Kluwer Academic, Boston, 2000.

[14] L. Xia, Y. Zhu, J. Yang, J. Ye, and Z. Gu, "Implementing a Thermal-Aware Scheduler in Linux Kernel on a Multi-Core Processor," The Computer Journal, pp. 895-903, 2010.

[15] A. Merkel and F. Bellosa, "Balancing power consumption in multiprocessor systems," In Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 (EuroSys '06), ACM, New York, NY, USA, pp. 403-414, 2006.

[16] R. Cochran and S. Reda, "Consistent runtime thermal prediction and control through workload phase detection," In Proceedings of the 47th Design Automation Conference (DAC '10), ACM, New York, NY, USA, pp. 62-67, 2010.

[17] J. Choi, C. Y. Cher, H. Franke, H. Hamann, A. Weger, and P. Bose, "Thermal-aware task scheduling at the system software level," In Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED '07). ACM, New York, NY, USA, pp. 213-218, 2007.

[18] E. K. Ardestani, F.-J Mesa-Martinez, and J. Renau, "Cooling solutions for processor Infrared Thermography," Semiconductor Thermal Measurement and Management Symposium," SEMI-THERM 26th Annual IEEE, pp. 187-190, 2010

[19] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration," SIGARCH Comput. Archit. News 34, pp. 78-88, 2006.

[20] C. Zhu, Z. Gu, L. Shang, R. P. Dick, R. Joseph, "Three-Dimensional Chip-Multiprocessor Run-Time Thermal Management," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.27, no.8, pp. 1479-1492, Aug, 2008.

[21] A. K. Coskun, R. Strong, D. M. Tullsen, and T. S. Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," In Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems (SIGMETRICS '09), ACM, New York, NY, USA, pp. 169-180, 2009.

[22] P. Chaparro, J. Gonzalez, G. Magklis, Q. Cai, A. Gonzalez, "Understanding the Thermal Implications of Multi-Core Architectures," IEEE Transactions on Parallel and Distributed Systems, vol.18, no.8, pp.1055-1065, Aug, 2007.

[23] A. Kumar, L. Shang, L. S. Peh, and N. K. Jha, “HybDTM: a coordinated hardware-software approach for dynamic thermal management,” In Proceedings of the 43rd annual Design Automation Conference (DAC '06). ACM, New York, NY, USA, pp. 548-553, 2006.

[24] D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” In proc. Int. Symp. High-Performance Comp. Architecture, pp. 171-182, 2001.

[25] K. Lee and K. Skadron, “Using performance counters for runtime temperature sensing in high performance processors,” In proc. Workshop HP-PAC, Int. Parallel and Distrib. Process. Symp., pp. 232.1, April, 2005.

[26] T. Y. Wang and C. C. P. Chen, “SPICE-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on modeling order reduction,” Proceedings. 5th International Symposium on Quality Electronic Design, pp. 357-362, 2004.

[27] S. Reda, R. J. Cochran, and A. N. Nowroz, "Improved Thermal Tracking for Processors Using Hard and Soft Sensor Allocation Techniques," IEEE Transactions on Computers, vol.60, no.6, pp.841-851, June, 2011.

[28] T. Y. Wang and C. C. P. Chen, "3-D Thermal-ADI: a linear-time chip level transient thermal simulator," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.21, no.12, pp. 1434- 1445, 2002.

[29] T. Y. Wang and C. C. P. Chen, "SPICE-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on modeling order reduction," Proceedings. 5th International Symposium on Quality Electronic Design, pp. 357- 362, 2004.

[30] Y. Zhan; S. S. Sapatnekar, "Fast computation of the temperature distribution in VLSI chips using the discrete cosine transform and table look-up," Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, vol.1, pp. 87-92, January, 2005.

[31] C. H. Lim, W. R. Daasch, G. Cai, "A thermal-aware superscalar microprocessor," Proceedings. International Symposium on Quality Electronic Design, pp. 517-522, 2002.

[32] M. Huang, J. Renau, S. M. Yoo, J. Torrellas, "A framework for dynamic energy efficiency and temperature management," Proceedings. 33rd Annual IEEE/ACM International Symposium on Microarchitecture, pp.202-213, 2000.

[33] H. Sanchez, B. Kuttanna, T. Olson, M. Alexander, G. Gerosa, R. Philip, J. Alvarez,

"Thermal management system for high performance PowerPCTM microprocessors," Compcon '97. Proceedings, IEEE , pp. 325-330, Feburary, 1997.

[34] K. Skadron, T. Abdelzaher, M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," Proceedings. Eighth International Symposium on High-Performance Computer Architecture, pp. 17-28, Feburary, 2002.

[35] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. Vetterling, Numerical recipes in C. Cambridge Univ. Press, Cambride, U.K, 1992.