

北京师范大学

硕士学位论文

论文题目：多核芯片的温敏低功耗调度研究

作 者： 闫佳琪

导 师： 骆祖莹 副教授

系别年级： 信息科学与技术学院

学 号： 201121210009

学科专业： 通信与信息系统

完成日期： 2014年 1月

北京师范大学研究生院

北京师范大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

关于论文使用授权的说明

学位论文作者完全了解北京师范大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京师范大学。学校有权保留并向国家有关部门或机构递交论文的复印件和电子版，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文（保密学位论文解密后执行此规定）。

保密论文注释：经本人申请，学校批准，本学位论文定为保密论文，密级： ，期限： 年
自 年 月 日起至 年 月 日止，解密后适用本授权书。

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名： _____ 日期： _____

导师签名： _____ 日期： _____

多核芯片的温敏低功耗调度研究

摘要

当前，面向复杂应用的高性能片上系统为了规避和减轻功耗墙问题，延续摩尔定律，采用了实时温度功耗管理与多核并行计算结构两种主要的技术手段。

实时功耗温度管理（DPTM）通过对任务的准确预测与合理调度，可以有效地降低片上系统的运行能耗与峰值温度。为了获得更好的DPTM调度效果，本文在第一部分提出了一种精确的组合式任务预测算法，一种任务调度算法VP-TALK，进而构建了一个完整的DPTM原型系统。为了对复杂任务进行精确地任务预测，本文DPTM系统先将复杂任务按频谱长短分类为随机/周期/趋势等三种成分，然后采用灰色模型/傅里叶模型/径向基函数（RBF）神经网络模型对这三种成分进行组合分析，以获得精确的预测效果；基于精确预测的任务负载量，本文所提出的VP-TALK算法可以计算出最优电压-频率对的理想值，进而选择出两组与理想值相邻的电压-频率对、以获得两个现实的工作状态，并考虑核心温度和任务实时性的条件，VP-TALK算法将任务负载分配到这两个工作状态，以获得最优的DPTM效果；最后基于机器学习方法，我们综合四种源算法、构建了一套完整的DPTM原型系统。

实验结果表明：本文系统的任务预测组合方法的平均误差仅为2.89%；在相同的设定峰值温度约束下，与已有调度算法的能耗值相比，尽管假设了更为敏感的功率-温度影响关系，但对于较高的工作负载率，本文所提出的VP-TALK调度算法仍能够获得平均14.33%的能耗降低；本文所提出DPTM原型系统可以获得接近于理想的能耗优化效果；

多核片上系统(MPSOC)的低功耗设计与实时功耗温度管理(DPTM)是目前重要的研究问题。本文采用了自下而上的建模方法对MPSOC结构级热分析方法进行了研究，提出了三种具有不同算法复杂度与精度的热分析方法：模块级方法BloTAM、核级方法CorTAM、考虑本核内模块相互影响的改良核级方法BiCorTAM，均具有简单、高效、与现有简化模型兼容、易于扩展、能够解决温度对漏电流的影响等优点。

实验数据表明：对核数较多MPSOC进行热分析的时候，CorTAM算法的复杂度低但精度也低，BloTAM算法的精度高但复杂度也高，而BiCorTAM则具有算法复杂度低和精度高的优点，其中局部热点的温度增量平均误差可以控制在3%以下，热分析的速度实现了

近50倍以上的分析加速，两者均是较为理想的结构级热分析方法；采用BiCorTM等增量式的热分析建模方法，可以快速进行用于降低热点温度的MPSOC布图规划过程。

关键词： 实时功耗管理；实时温度管理；任务调度；热分析；多核片上系统；布图规划

On Temperature and Power Management for MPSoC

ABSTRACT

To tackle the Power Wall problem on high performance chip processors oriented for heavy-load applications and continue Moore's Rule, Dynamic Power and Temperature Management (DPTM) and Multi-chip Processor Architecture are adopted in the field of Integrated Circuits (IC) design.

Optimal DPTM methods can effectively cut down the soaring power consumption and alleviate the problem of chip temperature. In order to get better scheduling results, this paper mainly accomplish three things. First, with principles derived from analyzing three previous methods as thumb rules, we obtain an improved DPTM algorithm, named VP-TALK, that carefully schedule the processor's running and dormant behaviors. Besides, we propose a combined predicting model. It may predict the workload on the chip so as to draw out optimal but unpractical frequency (F) and voltage (V). This F & V pair decides two distinct pairs of F & V, with which VP-TALK schedules the processor according to both the core temperature and remaindering work load. Finally, combining the workload prediction method and four DPTM algorithms, we further build a DPTM control system. Even though our model assume a tighter and more sensitive relationship between energy and temperature, experiments show that: 1) the workload prediction's error is as less as 2.89%; 2) under even more tough assumptions about thermal and power interrelation and the same peak temperature ceiling value, our proposed DPTM algorithm gains averagely 14.33% energy saving comparing to previous algorithms when the workload ratio is comparatively high; 3) comprehensive DPTM control system's managing effect is near to the most ideal one.

Efficient thermal analysis plays a key role in the temperature-aware floorplan design for MultiProcessor System-on-Chip (MPSoC) and DPTM. This paper adopts bottom-up modeling method to study architecture-level MPSoC thermal analysis method. First extract relative thermal resistance between functional modules with HotSpot software, then, based on these parameters, we propose three analysis methods with different accuracy and algorithm complexity: Block-level Temperature Analysis Method (BloTAM), Core-level Temperature Analysis Method (CorTAM)

and Block Improved Core Temperature Analysis Method (BiCorTAM). Experiment shows that BloTAM and BiCorTAM substantially reduce the time for MPSoC thermal analysis with guarantee of accuracy: speedup as high as 100 times is achieved with average temperature delta error as low as 3%. Both are ideal system-level analysis method.

KEY WORDS: Dynamic Power Management, Dynamic Temperature Management, Thermal Analysis, Task Scheduling, MPSoC, Architecture level

目 录

摘 要	I
ABSTRACT	III
1 引言	1
1.1 研究背景与相关科学问题	1
1.2 已有研究成果及其缺陷	1
1.3 本文工作及其贡献	2
1.4 文章结构安排	3
2 基于任务精确预测的实时功耗温度管理	4
2.1 实时系统的工作负载模型	4
2.2 实时系统的热分析模型	4
2.3 实时系统的功耗分析模型	4
2.4 已有的DPTM调度算法	5
2.4.1 TALK算法	5
2.4.2 Pattern-Based算法（简称PB算法）	5
2.4.3 M-Oscillating算法（简称MO算法）	6
2.4.4 对已有算法的评估	6
2.5 基于电压预测的TALK算法：VP-TALK	6
2.6 DPTM原型系统	9
2.6.1 启发性示例	9
2.6.2 基于机器学习的DPTM原型系统	10
2.6.3 基于单一调度策略的DPTM原型系统	12
3 DPTM原型系统模拟实验	13
3.1 模拟实验环境	13
3.2 VP-TALK算法的DPTM效果验证	13
3.3 基于机器学习的DPTM原型系统的验证	15
3.4 小结	18
4 考虑温度对漏电流功耗影响的MPSoC结构级热分析方法	19
4.1 多核芯片热分析研究基础	19
4.1.1 多核架构及其电热分布	19
4.1.2 芯片热分析及HotSpot模块级模型	20
4.1.3 电热耦合效应：温度对漏电流功耗的影响	22

4.2 3种MPSoC结构级热分析方法	23
4.2.1 模块级热分析方法BloTAM	23
4.2.2 核级热分析方法CorTAM	23
4.2.3 考虑核内模块相互影响的改良核级热分析方法BiCorTAM	23
参考文献	24
附录 1 外文资料原文	27
1.1 Single-Objective Programming	27
1.1.1 Linear Programming	28
1.1.2 Nonlinear Programming	29
附录 2 外文资料的调研阅读报告或书面翻译	30
2.1 单目标规划	30
2.1.1 线性规划	30
2.1.2 非线性规划	31
附录 3 其它附录	32
学术成果	33
致谢	34

1 引言

1.1 研究背景与相关科学问题

当前，面向复杂应用的高性能片上系统为了规避和减轻功耗墙（Power Wall）问题[1]，延续摩尔定律[2]，采用了两种主要的技术手段。

首先，必须在芯片运行中，通过合理任务调度来降低芯片的运行能耗和峰值工作温度。因此，对芯片进行实时功耗温度管理(DPTM)的算法研究就具有重要的理论意义与广阔的应用前景，是目前电子设计自动化(EDA)研究的一个热点问题。最初为了降低芯片运行功耗、延长设备电池的使用寿命，研究人员提出了运用动态电压调节技术（DVS）对系统动态功耗进行实时功耗管理（DPM）[3–6]。然而，随着IC进入纳米工艺，漏电流静态功耗已经超过动态功耗，成为芯片功耗的主要来源，而且漏电流和工作温度之间存在指数关系[7–9]，如对于65nm工艺，当温度从60摄氏度增加到80摄氏度，芯片漏电流会增加21%。

其次，目前IC业界已经普遍采用多核并行计算结构来提升芯片性能（通量）、降低设计复杂度。采用多核并行计算架构的多核片上系统(MPSOC)带来了热点分散的问题，即每个核都会产生一个局部热点[10]。为了将MPSOC多个热点的温度控制在一个安全阈值内，必须在设计与运行阶段，以功能模块与处理器核为单位，对芯片的功耗分布[11–13]与任务调度[14–16]进行优化，为此需要在结构级对芯片进行快速准确的热分析[13,14,17]。鉴于纳米工艺CMOS器件的漏电流随着工作温度的升高而指数增加，漏电流功耗与温度之间存在直接的依赖关系，即电热耦合效应[14]。为了提高分析的精度，必须在结构级热分析方法研究中考虑电热耦合效应[14–16]。

1.2 已有研究成果及其缺陷

针对任务调度领域，研究人员开始针对微处理器和大型服务器系统进行实时温度管理（DTM）[18–20]。为了对片上系统进行功耗、温度的统一调度与管理，最近开始出现了实时功耗温度管理(DPTM)的研究报道[21–24]，在考虑漏电流、温度相互作用关系和实时任务的时间限制这两个前提下，采用不同的DPTM策略来达到最小化运行能耗的目的。

在DPTM研究中，为了提高DPTM系统的降温降耗效果，必须对系统的任务负载进行精确的预测，事实上，任务负载的轻重决定了不同方法的DPTM效果。

对于多核芯片的热分析，受惠于电热分析的相似性，可以采用有限差分方法(PDF)可以进行全芯片三维热分析，获得温度分布的精确解[25]；为了考虑温度对功耗的影响，可以采用迭代方法来逼近最后的精确解[26]。基于PDF求解的HOTSPOT是目前广泛采用的热分析工具软件，能够用于MPSoC的结构级热分析，也能够对电热耦合效应进行求解[26]。尽管PDF方法可以获得高精度的求解方案，但这类方法的算法复杂度非常高，不满足MPSoC布图规划和实时功耗温度管理对结构级快速求解的需求[13,17]。为了对结构级设计的温度分布进行快速求解，出现过多种加速算法[10,11,14,15,17]。文献[11]采用最简单的物理距离模型，速度最快、精度最差，无法进行精确的MPSoC温度求解。文献[10,14,15]省略了核间的侧向热阻、来简化温度求解，其优点是速度快，缺点是降低了求解的精度。文献[27]采用基于学习的自回归算法进行在线温度分析，提高热分析速度的同时、也降低了求解的精度。总之，求解加速的代价是降低了求解的精度。为了考虑温度对功耗的影响（LDT），精确的求解算法必须采用迭代的方法进行逼近求解[13]。在现有结构级热分析算法中，为了提高求解速度，文献[10]没有考虑LDT，文献[15]采用线性模型来拟合LDT，文献[14,16]采用分段拟合系数矩阵来求解LDP效应，其结果会带来求解精度不同程度的降低。

1.3 本文工作及其贡献

为了弥补上文指出的已有研究的不足之处，本文对温度敏感的实时功耗调度和多核芯片的热分析方法这两个不同领域，分别做了较为深入的研究，并取得了如下成果。

首先，为了构建一个高效的DPTM系统，本文不仅提出了一种具有高精度的组合式任务预测方法，而且还提出一种新的DPTM任务调度算法VP-TALK，并进一步集成了一个基于负载预测的DPTM原型系统，该系统主要包括工作负载预测、任务实时调度两大模块。(1)基于组合任务预测方法的负载预测模块：根据频率范围，先将对应于复杂应用的任务分为随机/周期/趋势三种组分，然后采用灰色模型/傅里叶模型/RBF神经网络模型分别对这三种组分进行精确分析，最后将三部分预测结果合成为复杂任务的预测值。(2)基于多种调度算法的实时调度模块：先根据对工作负载率的精确预测值、计算出最优工作状态的电压/频率理想值，再从系统的电压/频率对的实际设定值中选取相邻的两个工作状态，最后考虑系统实时性、温度上限限制、静态功耗与温度的敏感关系以及芯片模式切换代价等多种因素，利用机器学习的方法，选择一种最优的调度策略。大量的模拟实验

表明, (1)在负载预测方面, 本文DPTM系统所采用的组合任务预测方法胜过众多的相关模型及算法, 平均误差仅为2.89%; (2)在节能效果方面, 当负载率高于55%时, 基于相同的峰值温度约束, 本文所提出的VP-TALK算法分别比Pattern-based、M-oscillating和TALK对比算法节能约20.5%、11.0%、11.5%; (3)本文DPTM原型系统的调度效果接近于理想调度效果。

其次, 本文采用自下而上的策略, 使用HOTSPOT提取MPSoC功能模块之间的热相关系数, 建立了模块级热分析方法BloTAM; 如图2所示, 每个核内只产生一个热点, 我们可以仅依靠热点之间的热相关系数、建立一个算法复杂度非常低的核级热分析方法CorTAM; 为了提高CorTAM的精度, 我们进一步提出了考虑本核内模块相互影响的改良核级方法BiCorTAM。与现有的结构级热分析算法相比, 本文所提出的三种方法均具有简单、高效、与现有简化模型兼容、易于扩展、考虑LDT影响等优点, 可以满足温敏MPSoC设计对高效、精确的结构级热分析方法的需求。与HOTSPOT软件的实验结果相比, 本文方法的实验数据表明: (1)对核数较多MPSoC进行局部热点温度分析的时候, BloTAM和BiCorTAM只产生2%、3%以下的温度增量平均误差, 是高精度的结构级热分析方法。 (2)在采用电压频率调节的温敏16核CPU布图规划研究中, 在包含参数提取时间的情况下, BloTAM和BiCorTAM可以提供50倍左右的计算加速。 (3)从总体效果来看, 在本文所提出三种建模分析方法中, BloTAM和BiCorTAM方法可以提供满意的分析精度与计算加速, 是较为理想的MPSoC结构级热分析方法。

1.4 文章结构安排

文章结构安排如下: 第二章介绍基于高精度组合式任务预测方法的DPTM原型系统, 研究对象仅限于单一处理器。第四章给出大量模拟实验数据, 以证实该调度系统在降低功耗和温敏控制上的优越性。第四章将研究对象扩展为多核处理器, 提出三种结构级热分析方法。为衡量这三种热分析计算模型的精确度与加速效果, 第五章中设计了若干实验测例, 并给出了模拟热分析结果。最后一章对全文作出总结。

2 基于任务精确预测的实时功耗温度管理

2.1 实时系统的工作负载模型

本文讨论的实时系统可以周期性地分配一段时间 D 作为执行某一任务的截止时间，该任务在最坏情况下所需要的执行时间为 W 。不失一般性，我们假设任务的截止时间等于系统周期性分配的时间片，并且等价地只考虑一个周期内的任务。文献[28]与[29]根据任务的性质研究了如何决定 (D, W) 数据对的值。本文中，由于可以预测出发送至实时系统的数据量，工作负荷便可以被认为是网络流量的归一化形式。

2.2 实时系统的热分析模型

为了研究处理器内核(Die)的热传导特性，文献[4,29,30]等都广泛采用了等效RC电路方法进行热分析建模，并采用如2-1进行工作温度的求解

$$\frac{dT}{dt} = \frac{P}{C_{th}} - \frac{T - T_{amb}}{R_{th}C_{th}} = \alpha P - \beta(T - T_{amb}) \quad (2-1)$$

2-1式中 T 和 T_{amb} 分别代表芯片的温度与环境温度， P 代表芯片在时刻 t 的功耗， R_{th} 与 C_{th} 分别为等效热阻与等效热容。

2.3 实时系统的功耗分析模型

处理器的系统状态可以分为工作状态和休眠状态：只有在工作状态下处理器才执行任务；否则，处理器将进入休眠状态以减少功耗并降低自身温度。工作状态下的功耗：

$$P_{active} = CV_{dd}^2 f + N_{gate} I_{leakage} V_{dd} \quad (2-2)$$

2-2中的第一项代表动态功耗，第二项代表静态功耗。当给定供电电压 V_{dd} 后，工作频率 f 为

$$f = \frac{(V_{dd} - V_t)^\mu}{(V_{dd} T_{max})^\eta} \times 4.2824 \times 10^{14} \approx C_1 V_{dd} \quad (2-3)$$

由于与工作电压成正比，我们可以使用 2-4计算动态功耗

$$P_{active} = C_2 V_{dd}^3 \quad (2-4)$$

通过HSPICE软件进行的曲线拟合，与温度、电压相关的漏电流可写为

$$I_{leakage} = I(V_0, T_0)(AT^2 \exp(\frac{\alpha V_{dd} + \beta}{T}) + B \exp(\gamma V_{dd} + \delta)) \quad (2-5)$$

2-5式中 $A, B, \alpha, \beta, \gamma, \delta, \mu, \eta$ 是经验参数，由生产工艺所决定（本文的模拟实验默认选择采用65nm的工艺参数）。当工作温度 T 在300K到380K的正常范围变化时， $\exp(\frac{1}{T})$ 的波动变化很小。当给定了 V_{dd} 后，文献[22]通过引入两个参考温度 TH 和 TL 进一步将漏电流简化为温度的二次函数。于是，与漏电流相关的静态功耗可以用 2-6式计算

$$P_{leakage} = N_{gate}(\hat{A}T^2 + \hat{B})V_{dd} \quad (2-6)$$

2-6中， $\hat{A} = \frac{I_{leakage}(TH, V_{dd}) - I_{leakage}(TL, V_{dd})}{TH^2 - TL^2}$ ， $\hat{B} = I_{leakage}(TH, V_{dd}) - \hat{A}TL^2$ 。此外，处理器的工作状态切换是通过改变工作电压来实现的，状态切换所带来的开销包括能耗开销 p_r 与延时开销 c_r [23]。整体而言，工作状态切换跨度越大，其能耗和时间的开销也就越大。

2.4 已有的DPTM调度算法

2.4.1 TALK算法

TALK及其改进算法[21]根据工作负载和截止时间的不同，来控制不同时间段处理器的工作/休息状态：当负载量大并且温度较低时、处理器处于激活工作状态；当负载量小并且温度较高时，处理器切换到睡眠状态以减小能耗，以降低温度。

2.4.2 Pattern-Based算法（简称PB算法）

PB算法将任务的截止时间或者运行周期 D 等分为 n 个时间片段，每段长 $\Delta = D/n$ ，采用PB算法的处理器将工作于特定规则的模式中[22]：执行 $\Delta = D/n$ 时间后便切入休眠模式，以减少功耗并降低温度。文献[22]与[30]证明：如果重复这种运行模式足够多次，处理器将达到温度的平衡值，并进入稳定状态，即每个周期的初始温度和结束温度将趋向于稳定值，以便于分析。

2.4.3 M-Oscillating算法（简称MO算法）

上面介绍的TALK算法和PB算法都要求处理器的工作速度要大于或者等于负载率 W/D 。文献[29]证明，如果采用两个最接近的速度完成分配给处理器的任务，那么相对于采用其他的工作速度组合，处于该速度组下处理器的温度是最优的。如果进一步地将这种两步策略应用在m个时间片中，不仅温度可以进一步优化，还可以将 D 时间内的总功耗表达为 m 的函数，而且必然存在能耗最小化的 m 值[24]。由于要考虑电压切换所付出的时间开销和能耗开销，[24]给出了 m 所具有确定的上限值*Ceil*。

2.4.4 对已有算法的评估

作为温敏调度算法，TALK参照剩余任务量与当前温度、来合理地调度任务。然而，简单的开关模式无法利用DVS技术，只能工作在固定速度。而且状态切换所导致的时间、能耗开销也是不可避免的。根据切换时间和能耗开销[23]，从全速工作转变为零电压将产生最大的能耗和时间开销。无论采用TALK还是PB算法，都要求处理器工作在大于 W/D 的速度上。大多数具有DVS或DVFS功能的实时系统通常只允许芯片的电压为若干离散值，根据负载率来调整电压工作档。这往往会导致芯片实际上工作高于任务所需的速度，不仅增加了近似与电压三次方成正比的动态功耗和与电压近似成正比的静态功耗，而且加速了温度的攀升，抬高了平衡态时的温度，进一步导致漏电流近似平方速度的增长。G.Quan等[24]提出的MO算法存在两个主要缺陷。首先，假设功率为温度的线性函数，使得峰值温度较PB有很大降低。其次是在实际应用中不能忽略低工作负载率情况：当 W/D 小于处理器支持的最低工作速度时，MO只能退化为PB，以防止不必要的功耗增加。

2.5 基于电压预测的TALK算法：VP-TALK

根据我们在DPTM领域的研究经验，可以获得如下关于DPTM改进的几点经验准则：

- (1) 必须考虑温度对静态功耗的影响。本文将功耗定为温度的二次函数。在温度限制下，DPTM系统最好具有温敏调控功能。
- (2) 如果能够准确预测负载量，确定任务所需的工作频率或者工作电压，就可以提前调度，更好的满足实时性。
- (3) 芯片电压选取。由于工作电压决定了运行速度，应该采用MO的电压选取方法，即使得芯片运行速度刚好满足工作负载的需求，以最大程度地利用DVFS技术来降低能耗。
- (4) 调度过程中，必须考虑电压切换（状态切换）所带来的额外能耗、时间开销。

基于以上观察，我们提出一种改良后的TALK算法，即具有电压选择的TALK算法，本文称为VP-TALK，其算法流程图如图 2.1所示。

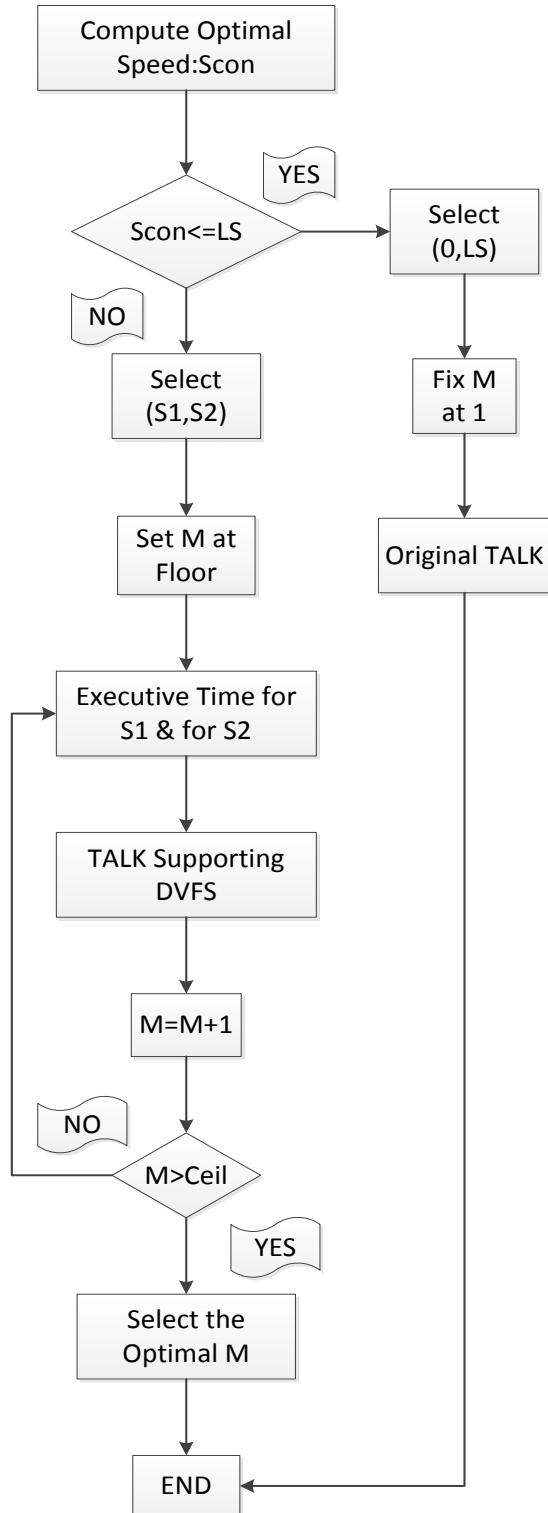


图 2.1 VP-TALK 算法流程

与MO相似，VP-TALK首先需要假设电压可以连续调节、以获得理论上的最优工作速度 S_{con} 。在 $S_{con} \leq LS$ （芯片最低工作速度）时，VP-TALK等同于原始的TALK算法。在 $S_{con} > LS$ 时，该算法选用两档邻接的速度 S_1 和 S_2 ，使得 $S_1 \leq S_{con} < S_2$ 。不同于MO的等分 M 段时间段，VP-TALK采用更灵活的、电压可调的温敏TALK来对每一小段的工作状态进行调度。由于 m 的数量由切换工作状态的代价和任务的实时性所限制，其上下限分别记为Ceil和Floor[24]。VP-TALK的应用前提是假设我们已经通过某种预测的方法预测出了任务负载量，从而，在任务到达前调度就已经开始，所以认为是实时性的。

2.6 DPTM原型系统

2.6.1 启发性示例

在之前的分析中，我们已经指出，较轻工作负载时的MO必然要退化至PB的方法。这是因为当工作所需电压低于可选的最小电压值时，MO中阶梯型工作电压策略无法通过逼近最优工作电压的方式节省动态功耗。为了探究工作量与最优调度算法之间的关系，我们设定任务长度为10秒，并考虑比文献[24]更强的温度对漏电流影响，我们在工作负载率全区间（5%-95%）范围内，对TALK、PB与MO这3种已有调度源算法的调度效果进行了考察。图2.2给出了能耗的数据，图2.3给出了温度数据。

我们可以得出如下结论：(1)当工作负载率(W/D)低于50%（近似值）时，三种源算法的峰值温度低于310K（37°C），峰值温度对系统的性能与可靠性没有影响；在系统能耗方面，PB算法（也即MO算法）的调度效果要胜过TALK算法，因此，PB算法具有最佳的调度效果。(2)当工作负载率(W/D)处于50%–70%区段（近似值）时，三种源算法的峰值温度低于320K（47°C），峰值温度对系统的性能与可靠性也没有影响；在系统能耗方面，TALK算法的调度效果要胜过PB和MO算法，因此，TALK算法具有最佳的调度效果。(3)当工作负载率(W/D)大于70%（近似值）时，三种源算法的峰值温度高于320K（47°C），最高可超过380K（107°C），峰值温度对系统的性能与可靠性具有明显影响，其中MO算法具有最低的峰值温度；在系统能耗方面，MO算法的调度效果要明显胜过其它两种源算法，MO算法具有最佳的调度效果。由此可见，最优的DPTM调度算法与工作负载率有直接关系。我们将以此关系作为理论基础，用于DPTM调度原型系统的构建，即根据对实时系统工作负荷的精确预测结果，来选择效果最佳的调度算法，并对DPTM调度效果进行评价。

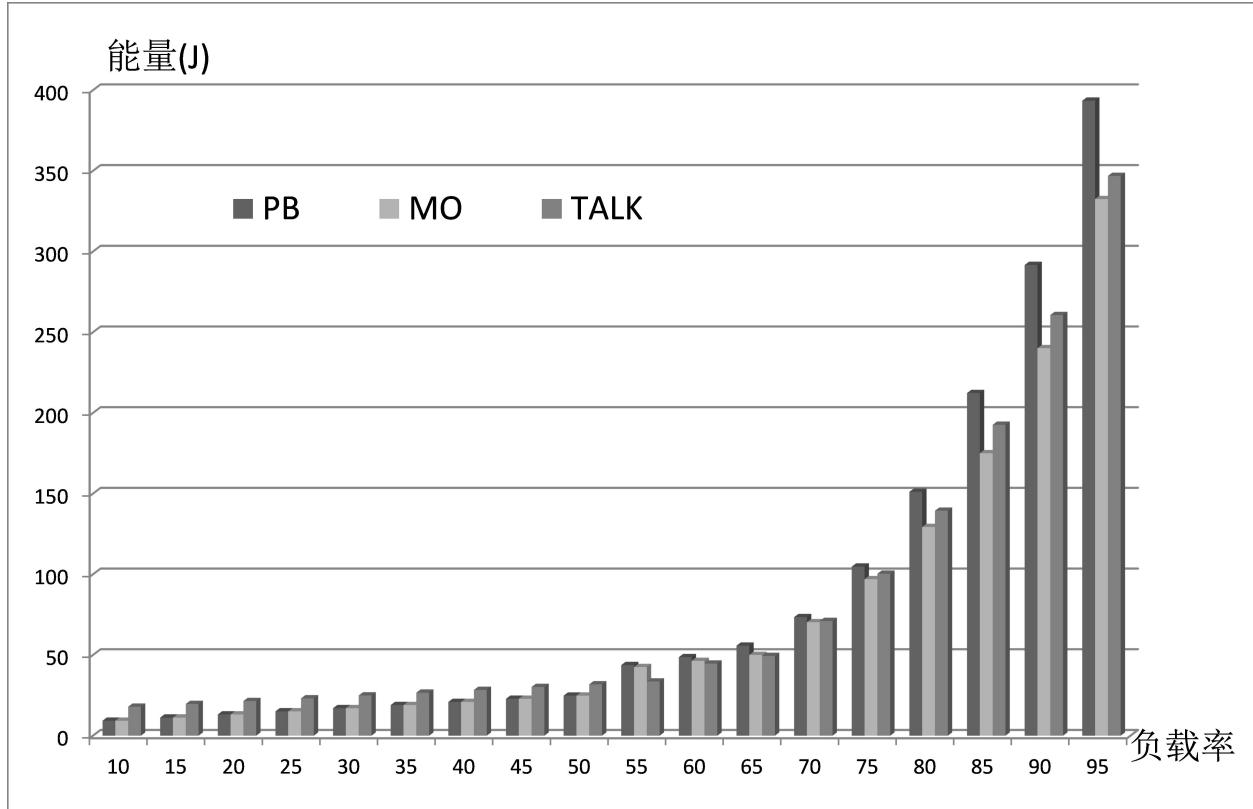


图 2.2 不同负载率下已有调度算法的能耗比较

2.6.2 基于机器学习的DPTM原型系统

根据调度算法性能与工作负载大小相关的观察，我们提出了基于工作负载预测结果来选择DPTM调度算法的调度策略，并以此构建了本文的DPTM原型系统，整个系统由工作负载预测、调度策略选择和调度策略评价三大模块组成，其具体的架构及其工作原理如图4所示。

在该系统工作中，其三大模块主要完成如下功能。（1）工作负载预测模块：我们根据负载变化周期的长短，提出了一种组合式的负载预测方法，采用多种不同拟合方法来分别对任务的不同物理意义成分进行精确预测，以获得对复杂任务的精确预测；（2）调度策略选择模块：我们综合考虑实时完成任务、温度上限、能耗最小化、漏电流与温度相关以及芯片模式切换代价等多种因素，选用不同的任务调度策略；（3）调度策略评价模块：对每种策略的系统能耗与峰值温度进行评价，并将其作为DPTM系统的反馈量，供调度策略选择模块参考。

调度策略选择的学习主要通过后期性能评价的评价值完成。假设存在 N 类DPTM，编号为 $1, 2, \dots, N$ 。它们在某一时刻 t 的得分或者权重为， $w_t = (w_{1t}, w_{2t}, \dots, w_{Nt})$ 对其中任一

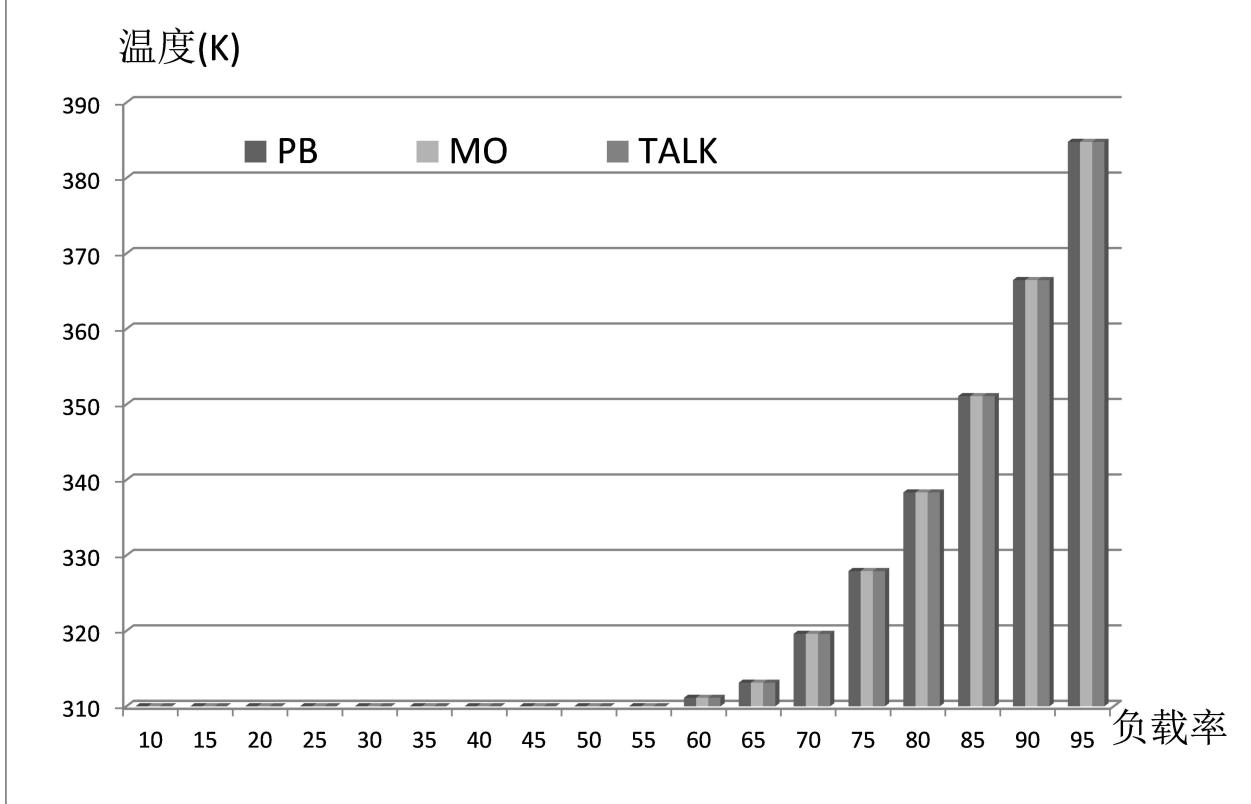


图 2.3 不同负载率下已有调度算法的温度比较

个权重分量 w_{kt} 可以采用 2-7 进行计算

$$w_{kt} = 1 - \sum_{j=t_0}^{t-1} \frac{E_{k,j} \lambda_j}{\sum_{i=1}^N E_{i,j}} \quad (2-7)$$

式中 $E_{i,j}$ 代表芯片使用第 i 类 DPTM 在时刻 j 的负载情况下消耗的能量， λ_j 为一可调整的参数，代表时刻 j 的能耗情况对决策的影响程度， t_0 是可以变动的初始值，它的取值意味着选取从何时开始的能耗情况作为以后决策的参考。在预测出工作负荷值后，则开始使用 2-8 进行决策：

$$DPTM_t = \arg(\max_{1 \leq k \leq N} (w_{k,t})) \quad (2-8)$$

2-8 中 $DPTM_t$ 为 t 时刻选择出的动态功耗温度调度策略。考虑到温度上的限制，我们需要考察所选的调度源算法是否会超过温度上限。如果能耗的节约是在很高的峰值温度的代价下换取的，我们将放弃该源算法，而选择次优的但是有较低峰值温度的调度源算法。与三种已有源算法和本文提出的 VP-TALK 算法相比，本文 DPTM 系统的主要扩展改进点在于：具有高精度的任务预测模块，为根据负载量而进行的策略选择提供前提基础；

通过基于调度效果评价的机器学习，自适应地根据负载量的轻重选择调度策略。

2.6.3 基于单一调度策略的DPTM原型系统

值得一提的是，如果我们选定某一种调度策略，省略机器学习模块，就构成了基于单一调度策略的DPTM原型系统。在该DPTM原型系统中，输入量为任务负载的历史值，通过这些历史值，利用第六节所述的任务负载预测模型，可以得到对于下一时刻任务量的预测值。进而，可以确定完成预测任务值所需要的芯片电压或者频率，并利用上文所述的某一种算法进行调度。不论是基于及其学习的DPTM系统，还是基于单一调度策略的DPTM系统，都要求提前预知任务负载的大小。因此，本文采用所提出的组合模型预测方法。该方法将复杂任务按频谱长短分类为随机/周期/趋势等三种成分，然后采用灰色模型/傅里叶模型/径向基函数（RBF）神经网络模型对这三种成分进行组合分析，可以得到平均相对误差低于3%、归一化方差小于0.5的任务负载预测效果[31]。

3 DPTM原型系统模拟实验

3.1 模拟实验环境

为了验证本文所提出的预测任务负载模型、VP-TALK调度算法以及DPTM原型系统，我们进行了三组模拟实验。实验运行平台为配有Intel Core 2 Q9550 CPU、4GB RAM的Windows7 64位操作系统，预测模型以及各DPTM算法在MATLAB[32]软件中模拟实现。

3.2 VP-TALK算法的DPTM效果验证

本文通过与现有的TALK、PB、MO算法进行对比，来验证本文提出的VP-TALK算法的优越性。实验中选取 p_r 、 c_r 取值分别为 $0.001s/V$ 和 $0.01J/V^2$ ，温度最高上限设为390K，任务周期固定为10s。支持DVFS芯片的电压值从0.6V到1.4V可调，步长0.1V。根据电压可以得到归一化的速度序列为0.574, 0.6611, 0.7324, 0.7926, 0.8446, 0.8901, 0.930, 0.9670, 1。任务负载率为5%至95%，变化步长为5%。对于4种对比算法，我们分别将能耗情况（单位为焦耳J）和峰值温度情况（单位为绝对温度K）绘制于图 3.1和图 3.2。

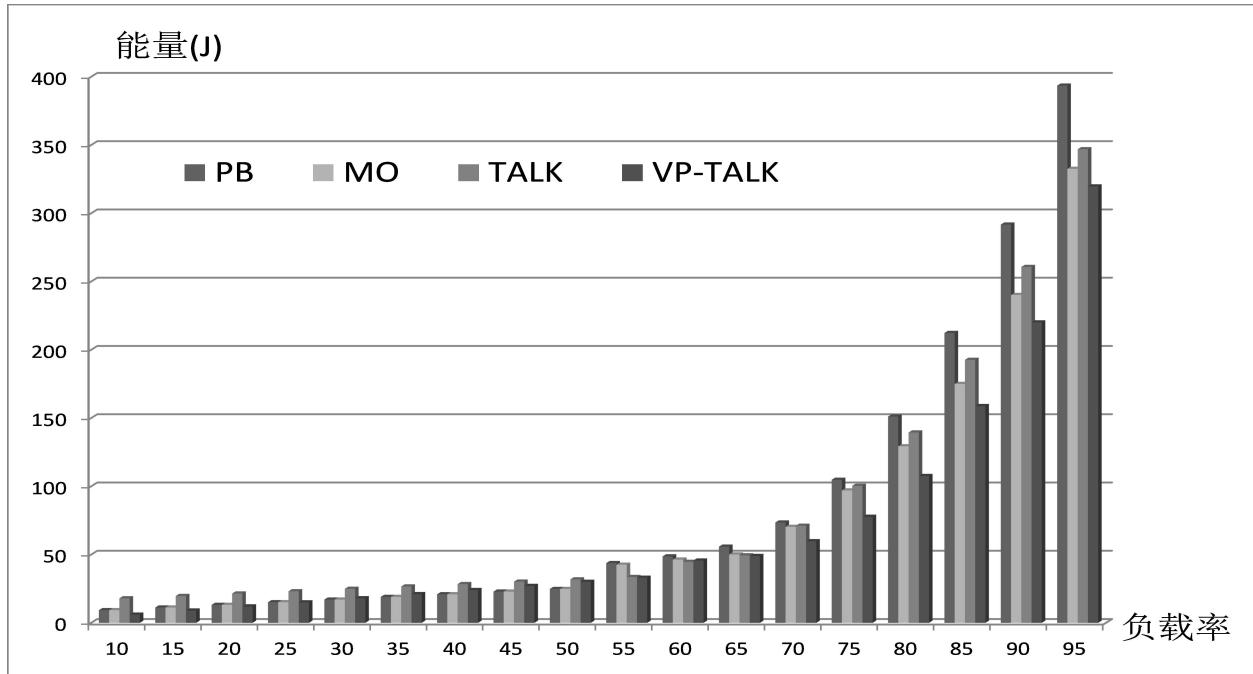


图 3.1 能耗比较

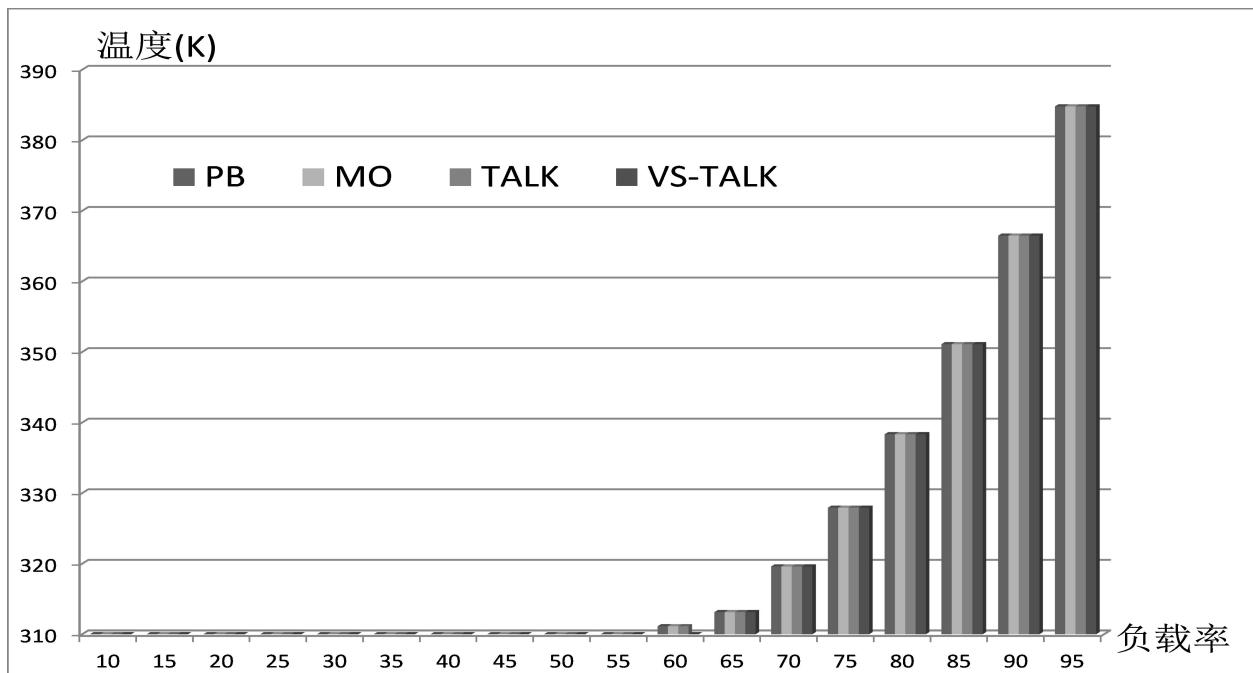


图 3.2 峰值温度比较

从图 3.1 可以看出，对于能耗：当负载率小于 55% 时，VP-TALK 优于 TALK、PB 与 MO 等价。这是因为，在负载率不高于芯片最低运行速度（0.574）

时，VP-TALK相当于使用DVS技术的TALK，PB相当于使用单一速度的MO。此时，TALK与PB的调度效果相差无几。在负载率为25%到50%时，PB略优于VP-TALK算法。原因是VP-TALK在电压和时间切换上付出了更大的代价。然而，随着负载率的增加，当负载率大于55%时，VP-TALK的调度效果全面优于其他三种算法，比PB、MO以及TALK的平均能耗分别节省大约20.54%、11.04%、11.42%。表2列出了当负载率大于55%时，能耗节省数据的统计信息，具体说明了VP-TALK的在能耗节约方面的优势。从图3.2中可以看出，当工作负载率小于50%时，各种算法所达到的峰值温度都不超过310K，四种调度算法在峰值温度方面的差距基本小于1K，并在最大负载率时近似共同终结于384K。

3.3 基于机器学习的DPTM原型系统的验证

为了评价本文基于机器学习的DPTM原型系统功耗与温度的调度效果，我们做了两组实验。首先，基于[31]中所得到的每个时刻工作负载预测值，我们分别采用四种对比源算法，即TALK[21]、PB[22]、MO[24]，VP-TALK，算出每个时刻每种算法的能耗值与峰值温度；然后将四种对比源算法每个时刻的能耗值与峰值温度进行平均，得到四种对比源算法每个时刻的平均调度效果；最后使用图3.3与图3.4分别比较了本文原型系统与四种对比源算法平均调度效果的能耗（单位J）和峰值温度（单位°C）。

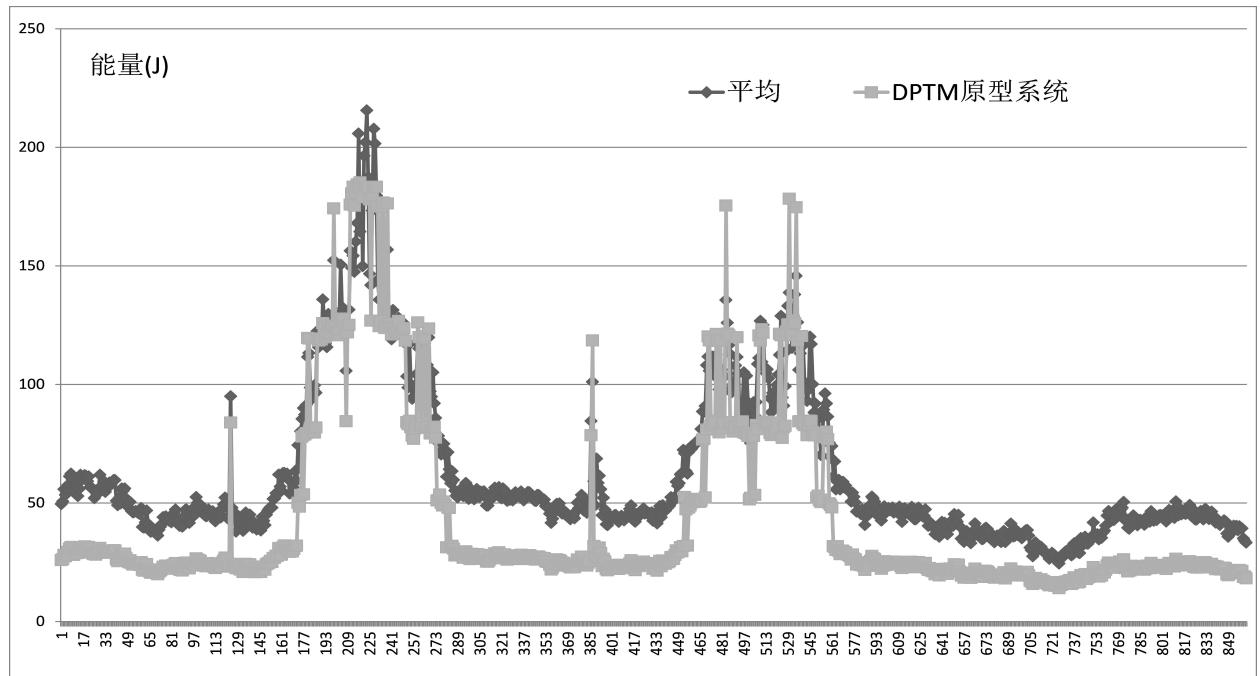


图3.3 DPTM原型系统VS源算法均值（能耗）

3 DPTM原型系统模拟实验

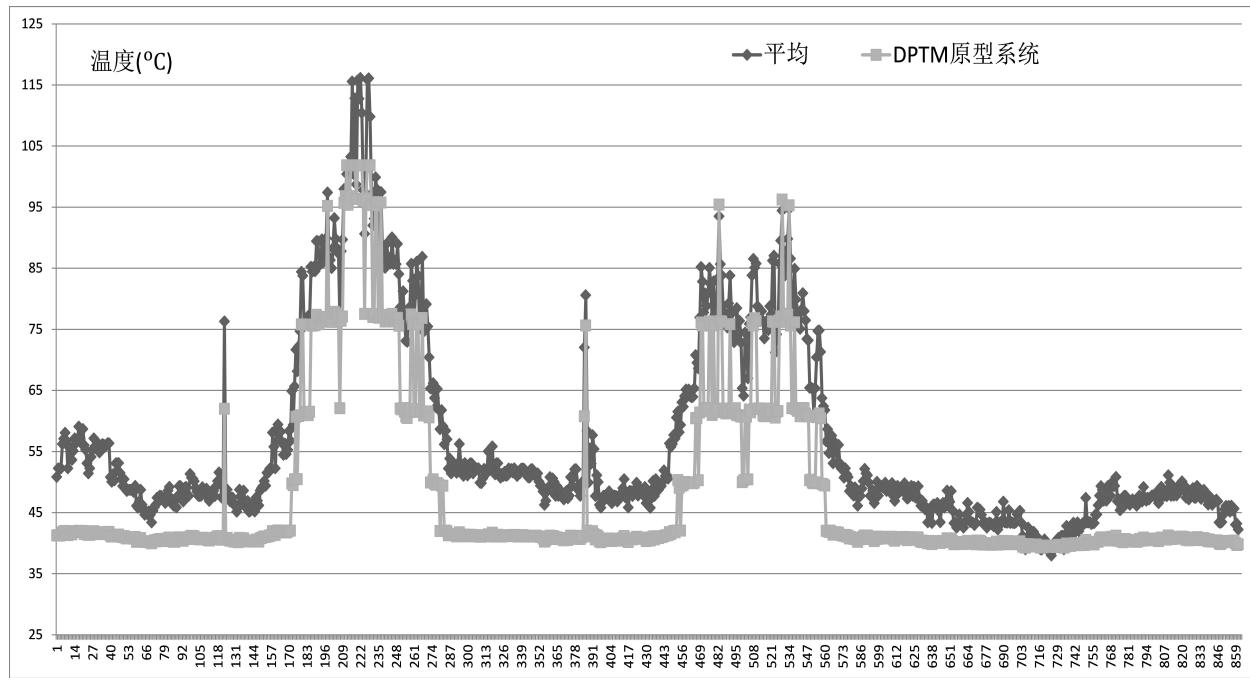


图 3.4 DPTM原型系统VS源算法均值 (峰值温度)

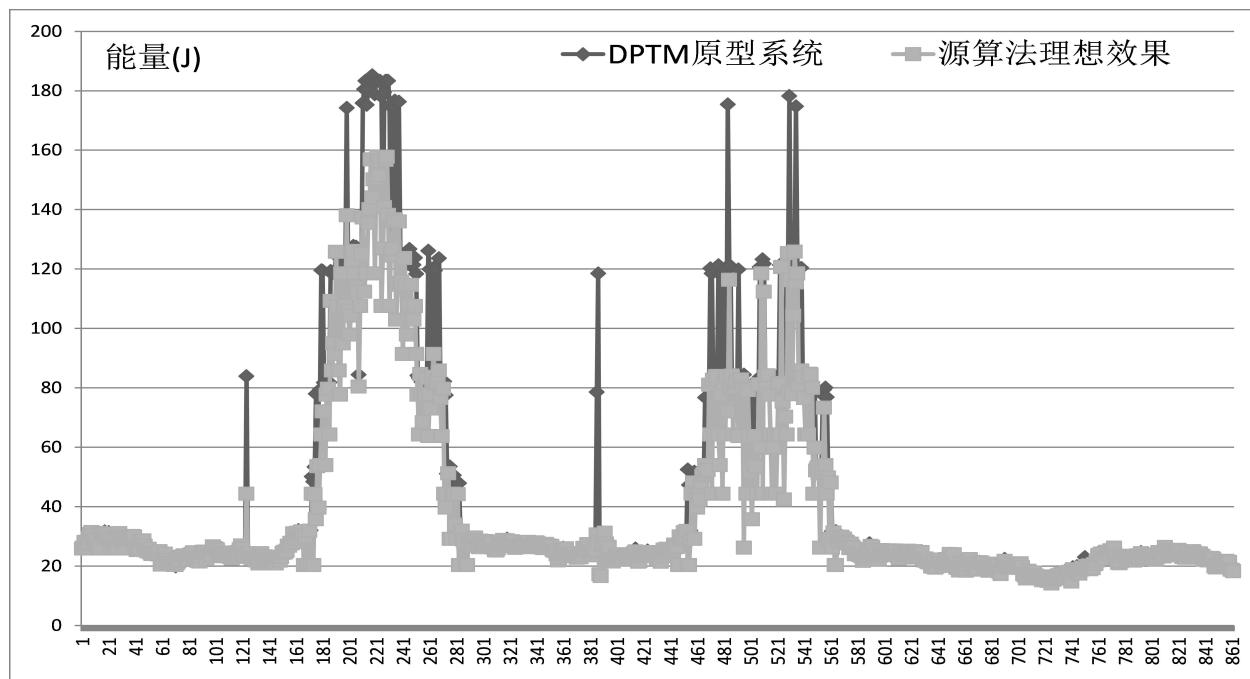


图 3.5 DPTM原型系统VS理想效果 (能耗)

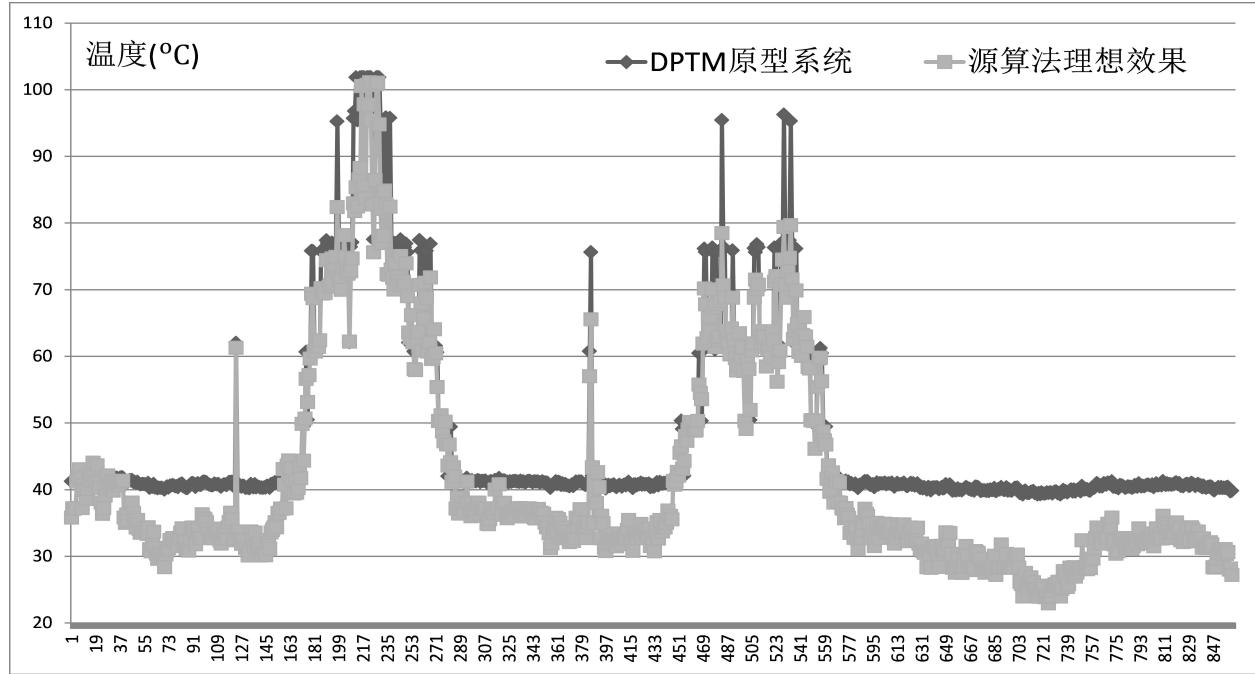


图 3.6 DPTM原型系统VS理想效果（峰值温度）

其次，同样基于[31]中所得到的每个时刻工作负载真实值，采用四种源算法，分别算出每个时刻的能耗值与峰值温度，取其中的最优值作为DPTM调度的理想值；然后使用这批理想值来客观评价集成了TALK、PB、MO以及VP-TALK算法的DPTM原型系统的调度效果；最后使用图 3.5与图 3.6分别比较了本文原型系统与四种对比源算法理论最优调度效果的能耗（单位J）和峰值温度（单位°C）。

从图 3.3与图 3.4可以直观地看出：与四种源算法平均调度效果相比，本文原型系统择优式的组合DPTM算法可以明显拉低实时系统的运行能耗曲线和峰值温度曲线，这表明对于四种DPTM源算法，本文基于机器学习的DPTM原型系统获得了“取其长、去其短”的优化效果。同时从表3的比较数据可以得出“本文原型系统可以获得近似最优的调度效果”的结论，其实验依据如下：(1)与四种DPTM源算法的平均效果相比（图 3.3），本文原型系统采用的择优式组合DPTM算法可获得更优的能耗优化效果，所有时间采样点能耗的累加值 E_{TOTAL} （总能耗）可获得18.39%的改进，所有时间采样点中的最大值 E_{MAX} （最大能耗）可获得18.77%的改进。整体改进效果非常明显。(2)与四种DPTM源算法相比（图 3.4），本文原型系统可以获得更优的峰值温度优化效果，所有时间采样点中的最大峰值温度 $T_{P_{MAX}}$ 是最优，所有时间采样点峰值温度的平均值 $T_{P_{AVG}}$ 是比算法均值稍弱。与四种DPTM源算法调度效果平均值相比，本文原型系统可以获得 $T_{P_{MAX}}$ 1.81°C的改进，但 $T_{P_{AVG}}$ 有-1.31°C的退化。从拉低最大峰值温度的角度来看，本文原型系统表现出了显著的优势。

度来讲，改进效果较为明显。(3)通过图3.5与图3.6对本文原型系统的调度效果与理想值进行的直观比较，看出本文原型系统可以获得比四种源算法均值更接近于理想值的优化效果。与表3中理想值的 $E_{TOTAL}/E_{MAX}/T_{PAVG}/T_{PMAZ}$ 参数相比，四种DPTM源算法调度效果平均值会产生28.64%/30.09%/7.71°C/9.73°C的差距，而本文原型系统只产生了12.55%/13.93%/9.02°C/7.91°C差距。

3.4 小结

第2章深入分析与评估了已有的主流调度算法，提出了一系列调度准则和经验。基于这些对芯片工作休眠状态调度的经验准则，我们提出一种在能耗节省方面更具优势的DPTM调度算法VP-TALK，以此算法为基础，综合本文所提出的预测任务负载模型，构建了一个基于负载预测的DPTM系统。本章中的仿真实验表明，(1)本文的组合模型在负载预测方面胜过众多的相关模型及算法，平均误差仅为2.89%；(2)本文所提出的VP-TALK算法在较高的工作负载率和共同的峰值温度约束下，分别比Pattern-Based、M-Oscillating、TALK分别节能20.54%、11.04%、11.42%；(3)本文所提出的综合四种源算法、基于机器学习的DPTM原型系统较为接近理想值，与其 $E_{TOTAL}/E_{MAX}/T_{PAVG}/T_{PMAZ}$ 参数相比，只产生了12.55%/13.93%/9.02°C/7.91°C差距。

4 考虑温度对漏电流功耗影响的MPSoC结构级热分析方法

4.1 多核芯片热分析研究基础

4.1.1 多核架构及其电热分布

目前多核CPU采用同质架构，即每个核拥有相同的逻辑功能模块、容量相同的专享缓存(cache)，占有相同的内核面积，同时共享最后一级缓存(last level cache,LLC)缓存、I/O等功能模块。每个核具有相同数量的工作模式，但不同的工作模式具有不同的能耗，即每个核具有一个全速高能模式外，还具有多个节能程度不同的节能模式。在每个核内，一般都具有一个功耗密度最大的逻辑功能模块、一个指令L1缓存和一个数据L1缓存，其功耗密度次之，一个功耗密度最小的L2缓存。由于注入的热量大，每个核的热点（温度最高点）出现在逻辑功能模块，所以在物理设计中，逻辑功能模块一般要布放在散热条件好的芯片边沿处，而将功耗密度最小的LLC布放在散热条件最差的芯片中央，以降低芯片的热点温度。图 4.1为Alpha 21264芯片的物理布局 [33]，

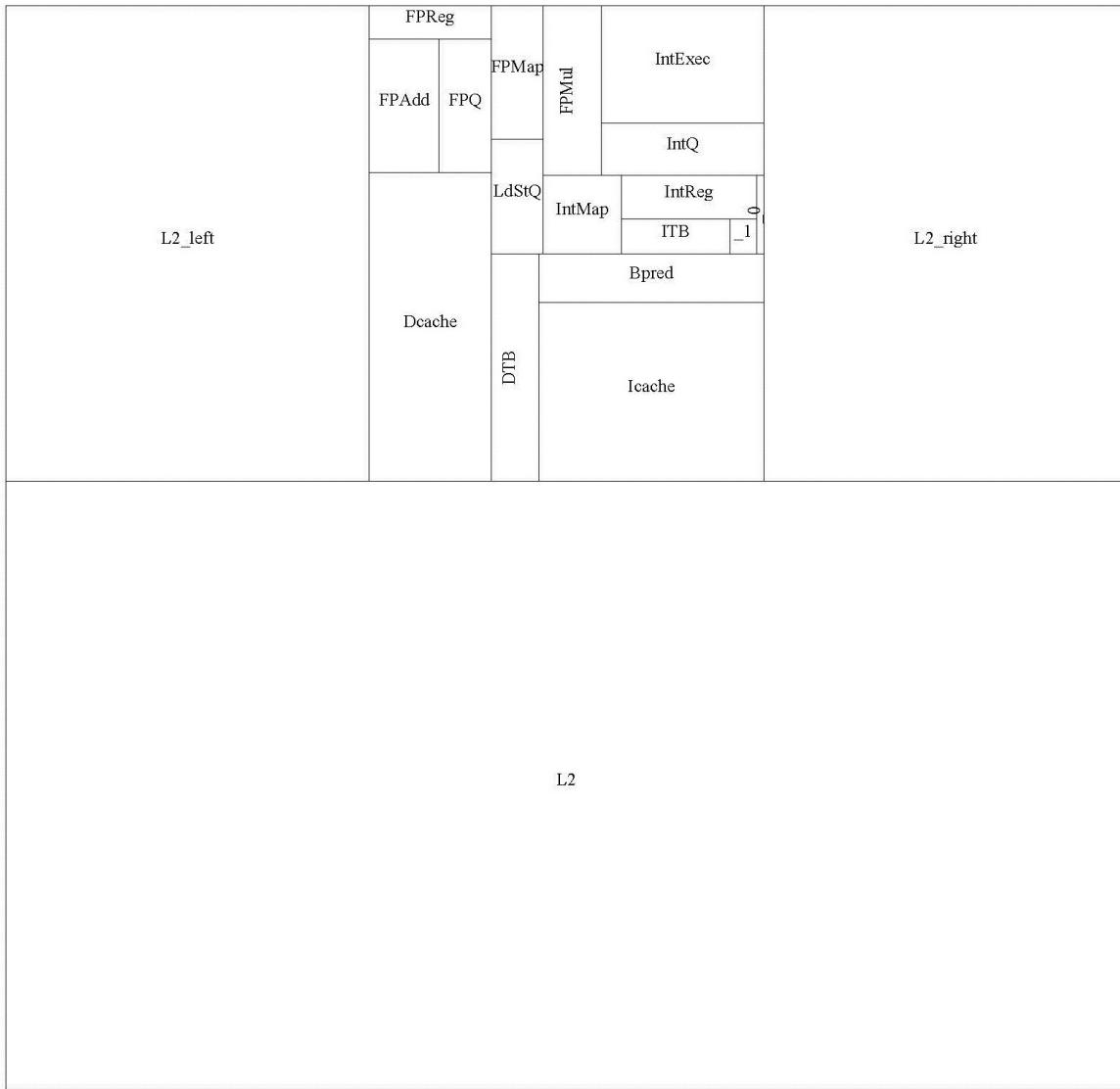


图 4.1 Alpha21264芯片的物理布局

4.1.2 芯片热分析及HotSpot模块级模型

在MPSoC结构级热分析中，一般采用稳态热分析方法计算温度分布，以降低计算复杂度[7,8]。对于稳态热分析而言，将芯片的功耗分布作为注入的热流向量 \mathbf{P} ，对芯片进行离散化建模后，可以获得节点之间的热导矩阵 \mathbf{G} ，目前多采用如下的稳态热分析方程计算节点温度分布向量 \mathbf{T} ：

$$\mathbf{G} \times \mathbf{T} = \mathbf{P} \quad (4-1)$$

对于多核实时功耗温度管理(dynamic power and temperature management, DPTM)研究[13, 34]，目前广泛采用Skadron等的HotSpot热分析模型[33]进行构建热导矩阵 \mathbf{G} ，并采用式4-1进行计算。HotSpot采用基于等效热导的电路模型，将体系结构级的功能块作为分析热点的对象。一种直观的对应芯片以及热封装的物理结构的具体建模例子如图4.2所示[14]。电路模型在垂直热传导方向上有3层：内核(die)层、扩热(heat spreader)层、与散热片(heatsink)层，另外加入第4层热对流(heat convector)层，即与环境温度的接口。内核层根据芯片的几何布局被分为块；扩热层分为5块：与内核层完好对应的Rsp以及4块呈梯形状的环绕块；散热片层按照与扩热层相似的划分方法，分为Rhs以及4个环绕块；最后，从热封装到外界环境的热对流层由Rconvection表示。层与层之间模型刻画由内核直至封装及外界环境的热流；层内水平模型刻画相邻模块间的热扩散。内核层产生的功耗等效为每个模块中心的电流源。建模完成后，通过电路分析可得到芯片的温度分布。用HotSpot计算分析Alpha 21264芯片的温度分布如图4.3所示。

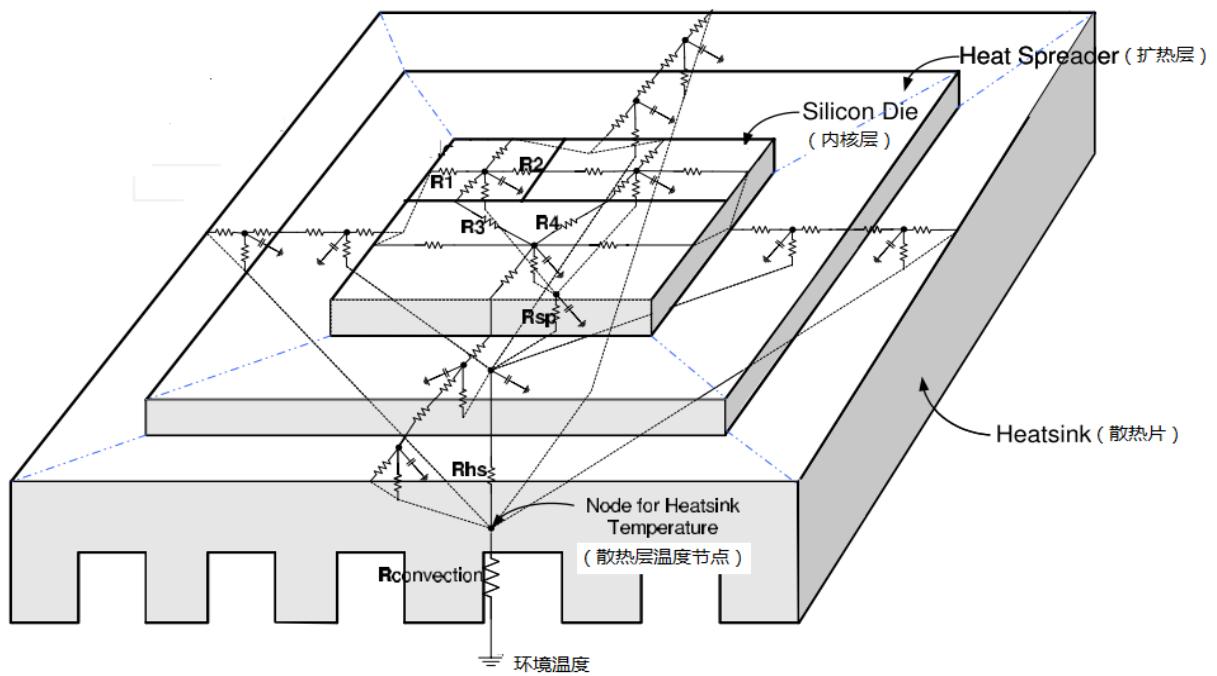


图 4.2 IC热分析的HotSpot分析模型

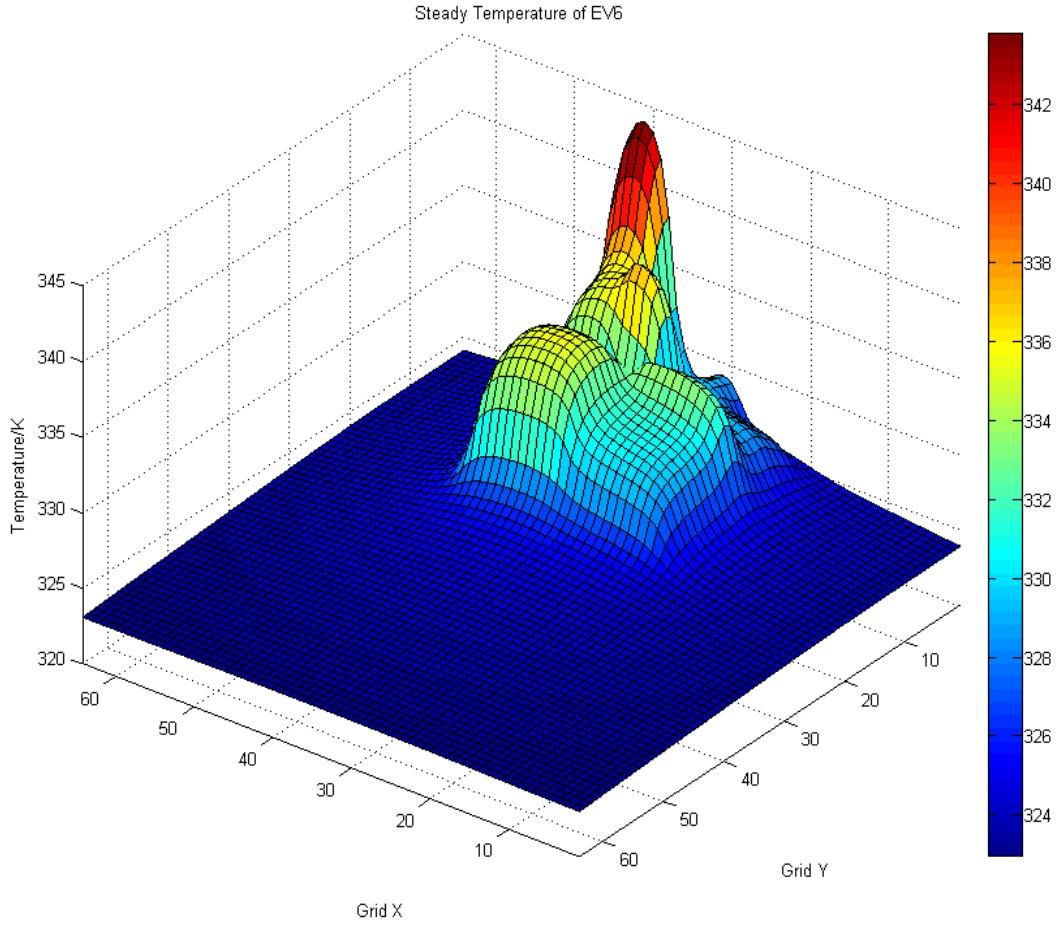


图 4.3 Alpha21264芯片的温度分布

4.1.3 电热耦合效应：温度对漏电流功耗的影响

正如 2.3 节中，式 2-4、2-6 所定性与定量地描述那样，芯片功耗由动态功耗 $P_{dynamic}$ 与静态功耗 $P_{leakage}$ 两部分组成，随着工艺的提高， $P_{leakage}$ 已成为芯片功耗的主要贡献者。而工作温度 T 的升高可以明显增大 $P_{leakage}$ ，此现象称之为电热耦合效应。考虑电热耦合效应的热分析过程中，需要对 $T - P$ 采用迭代计算。如 4.4 所示，对于一个 16 核 CPU 的测例（其具体的实验参数设置见本文后面的 3.1 部分），采用迭代算法来逼近最终的精确解，与不考虑电热耦合效应的初始解相比，芯片最高温度与静态功耗都有了明显的增加，这表明在芯片的温度分析中、必须考虑温度对静态功耗的影响，否则，将会产生较大的分析误差。同时，与不考虑电热耦合效应的温度分析算法相比，由于考虑电热耦合效应的温度分析算法需要采用 7 次迭代计算才能获得精确解，所以其算法复杂度是对比算法的 7 倍；

因此降低考虑电热耦合效应的温度分析算法复杂度就具有非常重要的研究意义

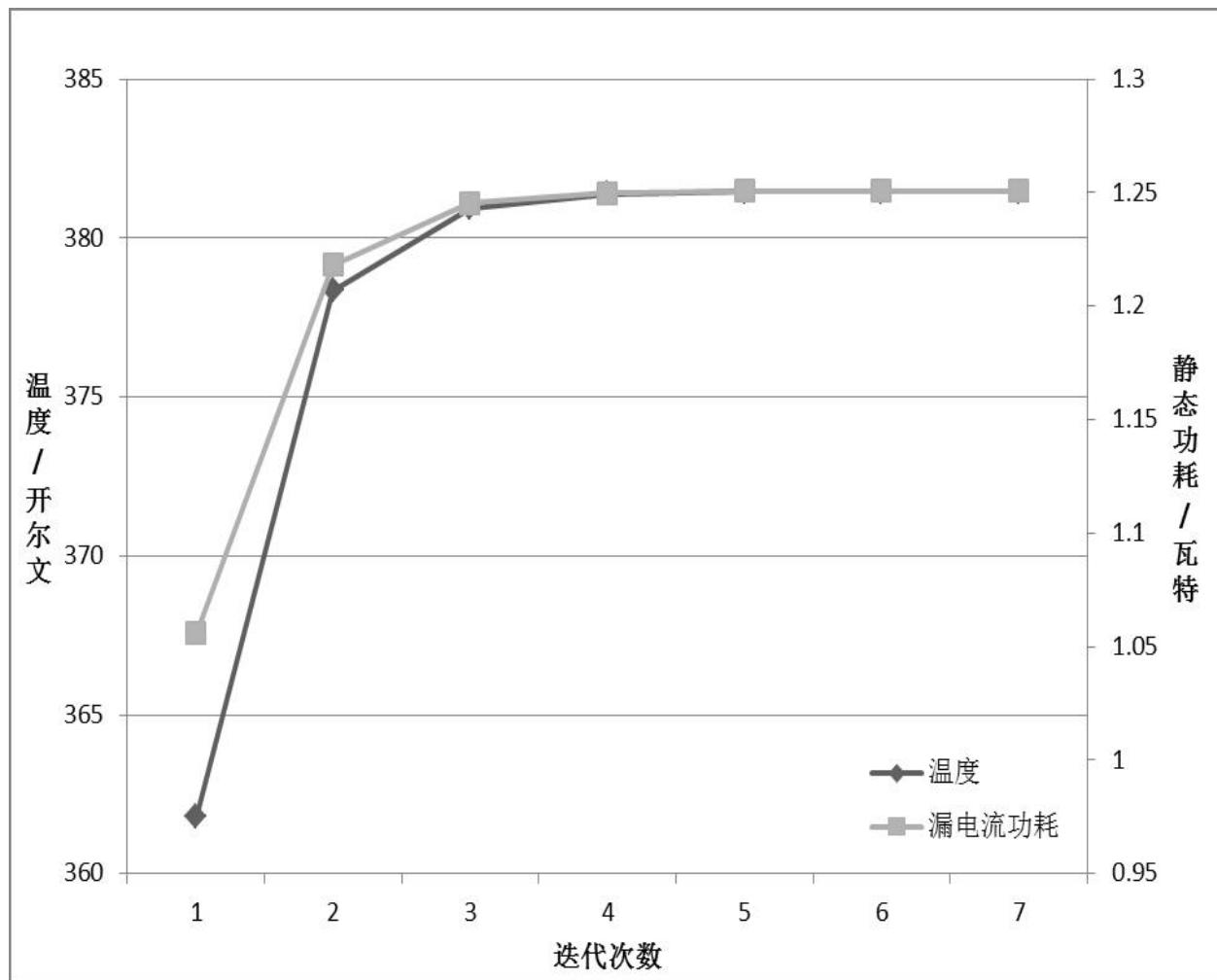


图 4.4 考虑电热耦合效应的多核芯片最高温度与静态功耗的迭代求解

4.2 3种MPSoC结构级热分析方法

4.2.1 模块级热分析方法BloTAM

4.2.2 核级热分析方法CorTAM

4.2.3 考虑核内模块相互影响的改良核级热分析方法BiCorTAM

参考文献

1. S. Borkar. Thousand Core Chips: A Technology Perspective[C]. Proceedings of the 44th Design Automation Conference, San Diego, USA, 2007. 746–749.
2. G. Moore. [C]. Proceedings of IEDM Tech Digest, 1975. 11–13.
3. B. Zhai. Theoretical and Practical Limits of Dynamic Voltage Scaling[C]. Proceedings of the 40th Design Automation Conference, San Diego, USA, 2004. 868–873.
4. R. Jejurikar, C. Pereira, R. Gupta. Leakage Aware Dynamic Voltage Scaling for Real-Time Embedded Systems[C]. Proceedings of the 40th Design Automation Conference, San Diego, USA, 2004. 275–280.
5. T. Chantem, R. P. Dick, X. S. Hu. Temperature-aware Scheduling and Assignment for Hard Real-time Applications on MPSoCs[C]. Proceedings of Design, Automation & Test in Europe, Munich, Germany, 2008. 288–293.
6. M. Santarini. Thermal Integrity: A Must for Low-power-IC Digital Design[J]. Sustainable Computing, Informatics and Systems, 2011, 4:286–293.
7. Z. G. Fu, C.S. Sun, Z. Y. Luo. A Task Scheduling Algorithm of Real-time Leakage Power and Temperature Optimization[C]. Proceedings of Computer Aided Design and Computer Graphics, Yellow Mountain City, China, 2009. 484–491.
8. Berkeley BSIM3 Device Models[Z]. [EB/OL]. <http://www.device.EECS.Berkeley.edu/bsim3>.
9. H. Sanchez, B. Kuttanna, T. Olson. Thermal Management System for High Performance Power PC Microprocessors[C]. Proceedings of International Conference on Technologies for the Information Superhighway, San Jose, USA, 1997. 325–330.
10. W. Huang, M. R. Stan, K. Sankaranarayanan. Many-core Design from a Thermal Perspective[C]. Proceedings of the 45th Design Automation Conference, Anaheim, California, USA: New York: ACM Press, 2008. 746–749.
11. M. B. Healy, H. H. S. Lee, G. H. Loh. Thermal Optimization in Multi-granularity Multi-core Floorplanning[C]. Proceedings of Asia and South Pacific Design Automation Conference. Piscataway, NJ: IEEE Press, 2009. 43–48.
12. K. Michael, R. Sherief. Frequency and Voltage Planning for Multi-core Processors under Thermal Constraints[C]. Proceedings of International Conference on Computer Design, Cancun, Mexico: Los Alamitos: IEEE Computer Society Press, 2008. 463–470.

13. K. Sankaranarayanan, B. H. Meyer, M. R. Stan. Thermal Benefit of Multi-core Floorplanning: A Limits Study[J]. Sustainable Computing, Informatics and Systems, 2011, 4:286–293.
14. V. Hanumaiah, R. Rao, S. Vrudhula. Throughput Optimal Task Allocation under Thermal Constraints for Multi-core Processors[C]. Proceedings of the 46th Design Automation Conference. New York: ACM Press, 2009. 776–781.
15. Y. Ge, Q. R. Qiu. Task Allocation for Minimum System Power in a Homogenous Multi-core Processor[C]. Proceedings of International Green Computing Conference. Los Alamitos: IEEE Computer Society Press, 2010.
16. C. L. Lung, Y. L. Ho, D. M. Kwai. Thermal-aware Online Task Allocation for 3D Multi-core Processor Throughput Optimization[C]. Proceedings of Design Automation & Test in Europe, Grenoble, France: New York: ACM Press, 2011. 1–6.
17. D. C. Juan, D. Marculescu. A Learning-based Autoregressive Model for Fast Transient Thermal Analysis of Chip-multiprocessors[C]. Proceedings of Asia and South Pacific Design Automation Conference, Sydney, Australia: Piscataway, NJ: IEEE Press, 2012. 597–602.
18. K Skadron, T. Abdelzaher, R. M. Stan, et al. Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management[C]. Proceedings of High Performance Computer Architecture, Boston, USA, 2002. 17–28.
19. K. Skadron, R. M. Stan, K. Sankaranarayanan. Temperature-aware Microarchitecture: Modeling and Implementation[J]. ACM Transactions on Architecture and Code Optimization, 2004, 1(1):94–125.
20. J Srinivasan, S. V. Adve. Predictive Dynamic Thermal Management for Multimedia Applications[C]. Proceedings of International Conference on Supercomputing, San Francisco, USA, 2003. 109–120.
21. L. Yuan, S. Leventhal, G. Qu. Temperature-aware Leakage Minimization Technique for Real-time Systems[C]. Proceedings of International Conference on Computer Aided Design, San Jose, USA: Piscataway, NJ: IEEE Press, 2006. 761–764.
22. C. Y. Yang, L. Thiele, T. W. Kuo. Energy-efficient Real-Time Task Scheduling with Temperature-Dependent Leakage[C]. Proceedings of International Conference on Computer Aided Design, Dresden, Germany, 2010. 9–14.
23. M. Bao, A. Andrei, P. Eles. Temperature-aware Idle Time Distribution for Energy Optimization with Dynamic Voltage Scaling[C]. Proceedings of Design, Automation & Test in Europe, Dresden, Germany, 2010. 21–27.
24. H. Huang, G. Quan. Leakage Aware Energy Minimization for Real-Time Systems under the Maximum Temperature Constraint[C]. Proceedings of Design, Automation & Test in Europe, Grenoble, France, 2011. 479–484.

25. Y. Zhan, B. Goplen, S. S. Sapatnekar. Electrothermal Analysis and Optimization Techniques for Nano-scale Integrated Circuits[C]. Proceedings of Asia and South Pacific Design Automation Conference, Yokohama, Japan: Piscataway, NJ: IEEE Press, 2006. 219–222.
26. W. Huang, S. Ghosh, S. Velusamy. HotSpot: A Compact Thermal Modeling Methodology for Early-stage VLSI Design[J]. IEEE Transactions on Very Large Scale Integration Systems, 2006, 14(5).
27. M. Janicki, J. H. Collet, A. Louri. HotSpots and Core-to-core Thermal Coupling in Future Multi-core Architectures[C]. Proceedings of the 26th IEEE Semiconductor Thermal Measurement and Management Symposium. Los Alamitos: IEEE Computer Society Press, 2010. 205–210.
28. L. Thiele, S. Chakraborty, M. Naedele. Real-time Calculus for Scheduling Hard Real-time Systems[C]. Proceedings of International Symposium on Circuits and System, Geneva, Switzerland, 2000. 101–104.
29. V. Chaturvedi, H. Huang, G. Quan. Leakage-aware Scheduling on Maximal Temperature Minimization for Periodic Hard Real-time Systems[C]. Proceedings of International Conference on Embedded Software and System, Bradford, UK, 2010. 1802–1809.
30. D. Rai, H. Yang, I. Bacivarov. Worst-Case Temperature Analysis for Real-Time Systems[C]. Proceedings of Design, Automation & Test in Europe, Grenoble, France, 2011. 631–636.
31. 闫佳琪, 骆祖莹, 赵国兴. 基于任务精确预测的实时功耗温度管理[C]. Proceedings of 第十七届全国计算机辅助设计与图形学学术会议暨第九届全国智能CAD与数字娱乐学术会议论文集. 清华大学出版社. 513–516.
32. MATLAB[Z]. [EB/OL]. <http://www.mathworks.com/products/matlab/>.
33. K. Skadron, M. R. Stan, W. Huang. Temperature-aware Microarchitecture[C]. Proceedings of International Symposium on Computer Architecture.
34. W. P. Liao, L. He, K. M. Lepak. Temperature and Supply Voltage Aware Performance and Power Modeling at Microarchitecture Level[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and System, 2005, 24(7):1042–1053.

附录 1 外文资料原文

As one of the most widely used techniques in operations research, *mathematical programming* is defined as a means of maximizing a quantity known as *objective function*, subject to a set of constraints represented by equations and inequalities. Some known subtopics of mathematical programming are linear programming, nonlinear programming, multiobjective programming, goal programming, dynamic programming, and multilevel programming^[1].

It is impossible to cover in a single chapter every concept of mathematical programming. This chapter introduces only the basic concepts and techniques of mathematical programming such that readers gain an understanding of them throughout the book^[2,3].

1.1 Single-Objective Programming

The general form of single-objective programming (SOP) is written as follows,

$$\begin{cases} \max f(x) \\ \text{subject to:} \\ g_j(x) \leq 0, \quad j = 1, 2, \dots, p \end{cases} \quad (123)$$

which maximizes a real-valued function f of $x = (x_1, x_2, \dots, x_n)$ subject to a set of constraints.

Definition 1.1: In SOP, we call x a decision vector, and x_1, x_2, \dots, x_n decision variables. The function f is called the objective function. The set

$$S = \left\{ x \in \Re^n \mid g_j(x) \leq 0, j = 1, 2, \dots, p \right\} \quad (456)$$

is called the feasible set. An element x in S is called a feasible solution.

Definition 1.2: A feasible solution x^* is called the optimal solution of SOP if and only if

$$f(x^*) \geq f(x) \quad (\text{附 1-1})$$

for any feasible solution x .

One of the outstanding contributions to mathematical programming was known as the Kuhn-Tucker conditions^{附 1-2}. In order to introduce them, let us give some definitions. An inequality

constraint $g_j(x) \leq 0$ is said to be active at a point x^* if $g_j(x^*) = 0$. A point x^* satisfying $g_j(x^*) \leq 0$ is said to be regular if the gradient vectors $\nabla g_j(x)$ of all active constraints are linearly independent.

Let x^* be a regular point of the constraints of SOP and assume that all the functions $f(x)$ and $g_j(x)$, $j = 1, 2, \dots, p$ are differentiable. If x^* is a local optimal solution, then there exist Lagrange multipliers λ_j , $j = 1, 2, \dots, p$ such that the following Kuhn-Tucker conditions hold,

$$\begin{cases} \nabla f(x^*) - \sum_{j=1}^p \lambda_j \nabla g_j(x^*) = 0 \\ \lambda_j g_j(x^*) = 0, \quad j = 1, 2, \dots, p \\ \lambda_j \geq 0, \quad j = 1, 2, \dots, p. \end{cases} \quad (\text{附 1-2})$$

If all the functions $f(x)$ and $g_j(x)$, $j = 1, 2, \dots, p$ are convex and differentiable, and the point x^* satisfies the Kuhn-Tucker conditions (附 1-2), then it has been proved that the point x^* is a global optimal solution of SOP.

1.1.1 Linear Programming

If the functions $f(x)$, $g_j(x)$, $j = 1, 2, \dots, p$ are all linear, then SOP is called a *linear programming*.

The feasible set of linear is always convex. A point x is called an extreme point of convex set S if $x \in S$ and x cannot be expressed as a convex combination of two points in S . It has been shown that the optimal solution to linear programming corresponds to an extreme point of its feasible set provided that the feasible set S is bounded. This fact is the basis of the *simplex algorithm* which was developed by Dantzig as a very efficient method for solving linear programming.

Table 1 This is an example for manually numbered table, which would not appear in the list of tables

Network Topology		# of nodes	# of clients			Server		
GT-ITM	Waxman Transit-Stub	600	2%	10%	50%	Max. Connectivity		
	Inet-2.1	6000						
Xue	Rui	Ni	BNUTHESIS					
	ABCDEF							

Roughly speaking, the simplex algorithm examines only the extreme points of the feasible set, rather than all feasible points. At first, the simplex algorithm selects an extreme point as the initial point. The successive extreme point is selected so as to improve the objective function

value. The procedure is repeated until no improvement in objective function value can be made. The last extreme point is the optimal solution.

1.1.2 Nonlinear Programming

If at least one of the functions $f(x), g_j(x), j = 1, 2, \dots, p$ is nonlinear, then SOP is called a *nonlinear programming*.

A large number of classical optimization methods have been developed to treat special-structural nonlinear programming based on the mathematical theory concerned with analyzing the structure of problems.

Now we consider a nonlinear programming which is confronted solely with maximizing a real-valued function with domain \Re^n . Whether derivatives are available or not, the usual strategy is first to select a point in \Re^n which is thought to be the most likely place where the maximum exists. If there is no information available on which to base such a selection, a point is chosen at random. From this first point an attempt is made to construct a sequence of points, each of which yields an improved objective function value over its predecessor. The next point to be added to the sequence is chosen by analyzing the behavior of the function at the previous points. This construction continues until some termination criterion is met. Methods based upon this strategy are called *ascent methods*, which can be classified as *direct methods*, *gradient methods*, and *Hessian methods* according to the information about the behavior of objective function f . Direct methods require only that the function can be evaluated at each point. Gradient methods require the evaluation of first derivatives of f . Hessian methods require the evaluation of second derivatives. In fact, there is no superior method for all problems. The efficiency of a method is very much dependent upon the objective function.

附录 2 外文资料的调研阅读报告或书面翻译

2.1 单目标规划

北冥有鱼，其名为鲲。鲲之大，不知其几千里也。化而为鸟，其名为鹏。鹏之背，不知其几千里也。怒而飞，其翼若垂天之云。是鸟也，海运则将徙于南冥。南冥者，天池也。

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x},y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \quad (123)$$

吾生也有涯，而知也无涯。以有涯随无涯，殆已！已而为知者，殆而已矣！为善无近名，为恶无近刑，缘督以为经，可以保身，可以全生，可以养亲，可以尽年。

2.1.1 线性规划

庖丁为文惠君解牛，手之所触，肩之所倚，足之所履，膝之所倚，砉然响然，奏刀騞然，莫不中音，合于桑林之舞，乃中经首之会。

表 1 这是手动编号但不出现在索引中的一个表格例子

Network Topology		# of nodes	# of clients			Server			
GT-ITM	Waxman Transit-Stub	600	2%	10%	50%	Max. Connectivity			
	Inet-2.1	6000							
Xue	Rui	Ni	BNUTHESIS						
	ABCDEF								

表 附 2.1 正常附录表格的例子

Network Topology		# of nodes	# of clients			Server			
GT-ITM	Waxman Transit-Stub	600	2%	10%	50%	Max. Connectivity			
	Inet-2.1	6000							
Xue	Rui	Ni	BNUTHESIS						
	ABCDEF								

文惠君曰：“嘻，善哉！技盖至此乎？”庖丁释刀对曰：“臣之所好者道也，进乎技矣。始臣之解牛之时，所见无非全牛者；三年之后，未尝见全牛也；方今之时，臣以神遇而不以目视，官知止而神欲行。依乎天理，批大郤，导大窾，因其固然。技经肯綮之未尝，而况大族乎！良庖岁更刀，割也；族庖月更刀，折也；今臣之刀十九年矣，所解数千牛矣，而刀刃若新发于硎。彼节者有间而刀刃者无厚，以无厚入有间，恢恢乎其于游刃必有余地矣。是以十九年而刀刃若新发于硎。虽然，每至于族，吾见其难为，怵然为戒，视为止，行为迟，动刀甚微，謋然已解，如土委地。提刀而立，为之而四顾，为之踌躇满志，善刀而藏之。”

文惠君曰：“善哉！吾闻庖丁之言，得养生焉。”

2.1.2 非线性规划

孔子与柳下季为友，柳下季之弟名曰盜跖。盜跖从卒九千人，横行天下，侵暴诸侯。穴室枢户，驱人牛马，取人妇女。贪得忘亲，不顾父母兄弟，不祭先祖。所过之邑，大国守城，小国入保，万民苦之。孔子谓柳下季曰：“夫为人父者，必能诏其子；为人兄者，必能教其弟。若父不能诏其子，兄不能教其弟，则无贵父子兄弟之亲矣。今先生，世之才士也，弟为盜跖，为天下害，而弗能教也，丘窃为先生羞之。丘请为先生往说之。”

柳下季曰：“先生言为人父者必能诏其子，为人兄者必能教其弟，若子不听父之诏，弟不受兄之教，虽今先生之辩，将奈之何哉？且跖之为人也，心如涌泉，意如飘风，强足以距敌，辩足以饰非。顺其心则喜，逆其心则怒，易辱人以言。先生必无往。”

孔子不听，颜回为驭，子贡为右，往见盜跖。

附录 3 其它附录

前面两个附录主要是给本科生做例子。其它附录的内容可以放到这里，当然如果你愿意，可以把这部分也放到独立的文件中，然后将其 \input 到主文件中。

学术成果

1. Jiaqi Yan, Zuying Luo, Liang Tang: 《Accurate Architecture-level Thermal Analysis Methods for MPSoC with Considering Leakage Power Dependence on Temperature》 ISQED2013: 178-183. (EI 收录, 检索号: 20132716473440)
2. 闫佳琪, 骆祖莹, 唐亮, 赵国兴: 《考虑温度对漏电流功耗影响的MPSoC结构级热分析方法》计算机辅助设计与图形学学报, 2013, 24 (11) : 1767-1774 (EI源刊)
3. 闫佳琪, 骆祖莹, 赵国兴: 《基于任务精确预测的实时功耗温度管理》 (已被计算机学报录用, EI源刊)

致 谢

衷心感谢导师骆祖莹副教授和赵国兴老师在研究生期间对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 xxx 教授热心指导与帮助，不胜感激。感谢 xx 实验室主任 xx 教授，以及实验室全体老师和同学们的热情帮助和支持！本课题承蒙国家自然科学基金资助，特此致谢。

感谢清华的薛瑞尼及相关同学，他们制作维护的清华学位论文模板极大的方便了 \LaTeX 用户的论文写作。

闫佳琪
2014年1月