

北京師範大學

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送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
1.5 手册	3
1.6 封面相关	4
1.7 字体命令	4
1.8 表格样本	5
1.8.1 基本表格	5
1.8.2 复杂表格	5
1.8.3 其它	9
1.9 定理环境	10
1.10 参考文献	13
1.11 公式	14
2 基于任务精确预测的实时功耗温度管理	15
2.1 实时系统的工作负载模型	15
2.2 实时系统的热分析模型	15
2.3 实时系统的功耗分析模型	15
2.4 已有的DPTM调度算法	16
2.4.1 TALK算法	16
2.4.2 Pattern-Based算法（简称PB算法）	16
2.4.3 M-Oscillating算法（简称MO算法）	16
2.4.4 对已有算法的评估	16
2.5 基于电压预测的TALK算法：VP-TALK	17
2.6 DPTM原型系统	18
2.6.1 启发性示例	18
2.6.2 基于机器学习的DPTM原型系统	18
2.6.3 基于单一调度策略的DPTM原型系统	19

2.7 其它例子	20
2.7.1 绘图	20
2.7.2 插图	20
2.7.2.1 一个图形	20
参考文献	22
附录 1 外文资料原文	26
1.1 Single-Objective Programming	26
1.1.1 Linear Programming	27
1.1.2 Nonlinear Programming	28
附录 2 外文资料的调研阅读报告或书面翻译	29
2.1 单目标规划	29
2.1.1 线性规划	29
2.1.2 非线性规划	30
附录 3 其它附录	31
学术成果	32
致谢	33

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。

1.5 手册

这是 BNUTHESES 的示例文档, 基本上覆盖了模板中所有格式的设置。建议大家在使用模板之前, 除了阅读《BNUTHESES用户手册》, 这个示例文档也最好能看一看。

小老鼠偷吃热凉粉；短长虫环绕矮高粱。^①

1.6 封面相关

封面的例子请参看 `cover.tex`。主要符号表参看 `denation.tex`，附录和个人简历分别参看 `appendix01.tex` 和 `resume.tex`。里面的命令都非常简单，一看即会。^②

1.7 字体命令

苏轼（1037-1101），北宋文学家、书画家。字子瞻，号东坡居士，眉州眉山（今属四川）人。苏洵子。嘉佑进士。神宗时曾任祠部员外郎，因反对王安石新法而求外职，任杭州通判，知密州、徐州、湖州。后以作诗“谤讟朝廷”罪贬黄州。哲宗时任翰林学士，曾出知杭州、颖州等，官至礼部尚书。后又贬谪惠州、儋州。北还后第二年病死常州。南宋时追谥文忠。与父洵弟辙，合称“三苏”。在政治上属于旧党，但也有改革弊政的要求。其文汪洋恣肆，明白畅达，为“唐宋八大家”之一。其诗清新豪健，善用夸张比喻，在艺术表现方面独具风格。少数诗篇也能反映民间疾苦，指责统治者的奢侈骄纵。词开豪放一派，对后代很有影响。《念奴娇·赤壁怀古》、《水调歌头·丙辰中秋》传诵甚广。

坡仙擅长行书、楷书，取法李邕、徐浩、颜真卿、杨凝式，而能自创新意。用笔丰腴跌宕，有天真烂漫之趣。与蔡襄、黄庭坚、米芾并称“宋四家”。能画竹，学文同，也喜作枯木怪石。论画主张“神似”，认为“论画以形似，见与儿童邻”；高度评价“诗中有画，画中有诗”的艺术造诣。诗文有《东坡七集》等。存世书迹有《答谢民师论文帖》、《祭黄几道文》、《前赤壁赋》、《黄州寒食诗帖》等。画迹有《枯木怪石图》、《竹石图》等。

易与天地准，故能弥纶天地之道。仰以观於天文，俯以察於地理，是故知幽明之故。原始反终，故知死生之说。精气为物，游魂为变，是故知鬼神之情状。与天地相似，故不违。知周乎万物，而道济天下，故不过。旁行而不流，乐天知命，故不忧。安土敦乎仁，故能爱。范围天地之化而不过，曲成万物而不遗，通乎昼夜之道而知，故神无方而易无体。

① 韩愈（768-824），字退之，河南河阳（今河南孟县）人，自称郡望昌黎，世称韩昌黎。幼孤贫刻苦好学，德宗贞元八年进士。曾任监察御史，因上疏请免关中赋役，贬为阳山县令。后随宰相裴度平定淮西迁刑部侍郎，又因上表谏迎佛骨，贬潮州刺史。做过吏部侍郎，死谥文公，故世称韩吏部、韩文公。是唐代古文运动领袖，与柳宗元合称韩柳。诗力求险怪新奇，雄浑重气势。

② 你说还是看不懂？怎么会呢？

有天地，然后万物生焉。盈天地之间者，唯万物，故受之以屯；屯者盈也，屯者物之始生也。物生必蒙，故受之以蒙；蒙者蒙也，物之穉也。物穉不可不养也，故受之以需；需者饮食之道也。饮食必有讼，故受之以讼。讼必有众起，故受之以师；师者众也。众必有所比，故受之以比；比者比也。比必有所畜也，故受之以小畜。物畜然后有礼，故受之以履。

履而泰，然后安，故受之以泰；泰者通也。物不可以终通，故受之以否。物不可以终否，故受之以同人。与人同者，物必归焉，故受之以大有。有大者不可以盈，故受之以谦。有大而能谦，必豫，故受之以豫。豫必有随，故受之以随。以喜随人者，必有事，故受之以蛊；蛊者事也。

有事而后大可，故受之以临；临者大也。物大然后可观，故受之以观。可观而后有所合，故受之以噬嗑；嗑者合也。物不可以苟合而已，故受之以贲；贲者饰也。致饰然后亨，则尽矣，故受之以剥；剥者剥也。物不可以终尽，剥穷上反下，故受之以复。复则不妄矣，故受之以无妄。

有无妄然后可畜，故受之以大畜。物畜然后可养，故受之以颐；颐者养也。不养则不可动，故受之以大过。物不可以终过，故受之以坎；坎者陷也。陷必有所丽，故受之以离；离者丽也。

1.8 表格样本

1.8.1 基本表格

模板中关于表格的宏包有三个：`booktabs`、`array` 和 `longtabular`，命令有一个 `\hlinewd`。三线表可以用 `booktabs` 提供的 `\toprule`、`\midrule` 和 `\bottomrule`。它们与 `longtable` 能很好的配合使用。如果表格比较简单的话可以直接用命令 `\hlinewd{xpt}` 控制。

首先来看一个最简单的表格。表 1.1 列举了本模板主要文件及其功能。请大家注意三线表中各条线对应的命令。这个例子还展示了如何在表格中正确使用脚注。由于 \LaTeX 本身不支持在表格中使用 `\footnote`，所以我们不得不将表格放在小页中，而且最好将表格的宽度设置为小页的宽度，这样脚注看起来才更美观。

1.8.2 复杂表格

我们经常会在表格下方标注数据来源，或者对表格里面的条目进行解释。前面的脚注是一种不错的方法，如果你不喜欢脚注。那么完全可以在表格后面自己写注释，比如

表 1.1 模板文件。如果表格的标题很长，那么在表格索引中就会很不美观，所以要像 `chapter` 那样在前面用中括号写一个简短的标题。这个标题会出现在索引中。

文件名	描述
<code>bnuthesis.ins</code>	L ^A T _E X 安装文件， <code>docstrip</code> ^①
<code>bnuthesis.dtx</code>	所有的一切都在这里面 ^② 。
<code>bnuthesis.cls</code>	模板类文件。
<code>bnuthesis.cfg</code>	模板配置文。 <code>cls</code> 和 <code>cfg</code> 由前两个文件生成。
<code>bnubib.bst</code>	参考文献 Bibtex 样式文件。
<code>bnutils.sty</code>	常用的包和命令写在这里，减轻主文件的负担。

① 表格中的脚注

② 再来一个

表 1.2。

表 1.2 复杂表格示例 1

	First Half		Second Half	
	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
East*	20.4	27.4	90	20.4
West**	30.6	38.6	34.6	31.6

注：数据来源《BNUTHESES 使用手册》。

*：东部

**：西部

此外，表 1.2 同时还演示了通过 `tabularx` 的 `|X|` 扩展实现表格自动放大；

为了使我们的例子更接近实际情况，我会在必要的时候插入一些“无关”文字，以免太多图表同时出现，导致排版效果不太理想。第一个出场的当然是我的最爱：风流潇洒、骏马绝尘、健笔凌云的**李太白**了。

李白，字太白，陇西成纪人。凉武昭王暕九世孙。或曰山东人，或曰蜀人。白少有逸才，志气宏放，飘然有超世之心。初隐岷山，益州长史苏颋见而异之，曰：“是子天才英特，可比相如。”天宝初，至长安，往见贺知章。知章见其文，叹曰：“子谪仙人也。”言于明皇，召见金銮殿，奏颂一篇。帝赐食，亲为调羹，有诏供奉翰林。白犹与酒徒饮于市，帝坐沉香亭子，意有所感，欲得白为乐章，召入，而白已醉。左右以水颡面，稍解，援笔成文，婉丽精切。帝爱其才，数宴见。白常侍帝，醉，使高力士脱靴。力士素贵，耻之，摘其诗以激杨贵妃。帝欲官白，妃辄沮止。白自知不为亲近所容，悬

求还山。帝赐金放还。乃浪迹江湖，终日沉饮。永王璘都督江陵，辟为僚佐。璘谋乱，兵败，白坐长流夜郎，会赦得还。族人阳冰为当涂令，白往依之。代宗立，以左拾遗召，而白已卒。文宗时，诏以白歌诗、裴旻剑舞、张旭草书为三绝云。集三十卷。今编诗二十五卷。

——《全唐诗》诗人小传

浮动体的并排放置一般有两种情况：1) 二者没有关系，为两个独立的浮动体；2) 二者隶属于同一个浮动体。对表格来说并排表格既可以像图 1.3、图 1.4 使用小页环境，也可以如图 1.5 使用子表格来做。

表 1.3 第一个并排子表格

111	222
222	333

表 1.4 第二个并排子表格

111	222
222	333

然后就是忧国忧民，诗家楷模杜工部了。杜甫，字子美，其先襄阳人，曾祖依艺为巩令，因居巩。甫天宝初应进士，不第。后献《三大礼赋》，明皇奇之，召试文章，授京兆府兵曹参军。安禄山陷京师，肃宗即位灵武，甫自贼中遁赴行在，拜左拾遗。以论救房琯，出为华州司功参军。关辅饥乱，寓居同州同谷县，身自负薪采相，餽糒不给。久之，召补京兆府功曹，道阻不赴。严武镇成都，奏为参谋、检校工部员外郎，赐绯。武与甫世旧，待遇甚厚。乃于成都浣花里种竹植树，枕江结庐，纵酒啸歌其中。武卒，甫无所依，乃之东蜀就高适。既至而适卒。是岁，蜀帅相攻杀，蜀大扰。甫携家避乱荆楚，扁舟下峡，未维舟而江陵亦乱。乃溯沿湘流，游衡山，寓居耒阳。卒年五十九。元和中，归葬偃师首阳山，元稹志其墓。天宝间，甫与李白齐名，时称李杜。然元稹之言曰：“李白壮浪纵恣，摆去拘束，诚亦差肩子美矣。至若铺陈终始，排比声韵，大或千言，次犹数百，词气豪迈，而风调清深，属对律切，而脱弃凡近，则李尚不能历其藩翰，况堂奥乎。”白居易亦云：“杜诗贯穿古今，尽工尽善，殆过于李。”元、白之论如此。盖其出处劳佚，喜乐悲愤，好贤恶恶，一见于诗。而又以忠君忧国、伤时念乱为本旨。读其诗可以知其世，故当时谓之“诗史”。旧集诗文共六十卷，今编诗十九卷。

表 1.5 并排子表格

(a) 第一个子表格

111	222
222	333

(b) 第二个子表格

111	222
222	333

不可否认 \LaTeX 的表格功能没有想象中的那么强大，不过只要你足够认真，足够细致，那么同样可以排出来非常复杂非常漂亮的表格。请参看表 1.6。

表 1.6 复杂表格示例 2

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

最后就是清新飘逸、文约意赅、空谷绝响的王大侠了。王维，字摩诘，河东人。工书画，与弟缙俱有俊才。开元九年，进士擢第，调太乐丞。坐累为济州司仓参军，历右拾遗、监察御史、左补阙、库部郎中，拜吏部郎中。天宝末，为给事中。安禄山陷两都，维为贼所得，服药阳喑，拘于菩提寺。禄山宴凝碧池，维潜赋诗悲悼，闻于行在。贼平，陷贼官三等定罪，特原之，责授太子中允，迁中庶子、中书舍人。复拜给事中，转尚书右丞。维以诗名盛于开元、天宝间，宁薛诸王驸马豪贵之门，无不拂席迎之。得宋之问辋川别墅，山水绝胜，与道友裴迪，浮舟往来，弹琴赋诗，啸咏终日。笃于奉佛，晚年长斋禅诵。一日，忽索笔作书数纸，别弟缙及平生亲故，舍笔而卒。赠秘书监。宝应中，代宗问缙：“朕常于诸王坐闻维乐章，今存几何？”缙集诗六卷，文四卷，表上之。敕答云，卿伯氏位列先朝，名高希代。抗行周雅，长揖楚辞。诗家者流，时论归美。克成编录，叹息良深。殷璠谓维诗词秀调雅，意新理惬。在泉成珠，著壁成绘。苏轼亦云：“维诗中有画，画中有诗也。”今编诗四卷。

要想用好论文模板还是得提前学习一些 $\text{\TeX}/\text{\LaTeX}$ 的相关知识，具备一些基本能力，掌握一些常见技巧，否则一旦遇到问题还真是比较麻烦。我们见过很多这样的同学，一直以来都是使用 Word 等字处理工具，以为 \LaTeX 模板的用法也应该类似，所以就沿袭同样的思路来对待这种所见非所得的排版工具，结果被折腾的焦头烂额，疲惫不堪。

如果您要排版的表格长度超过一页，那么推荐使用 `longtable` 或者 `supertabular` 宏包，模板对 `longtable` 进行了相应的设置，所以用起来可能简单一些。表 1.7 就是 `longtable` 的简单示例。

表 1.7 实验数据

测试程序	正常运行 时间 (s)	同步 时间 (s)	检查点 时间 (s)	卷回恢复 时间 (s)	进程迁移 时间 (s)	检查点 文件 (KB)
CG.A.2	23.05	0.002	0.116	0.035	0.589	32491
CG.A.4	15.06	0.003	0.067	0.021	0.351	18211
CG.A.8	13.38	0.004	0.072	0.023	0.210	9890
CG.B.2	867.45	0.002	0.864	0.232	3.256	228562
CG.B.4	501.61	0.003	0.438	0.136	2.075	123862
CG.B.8	384.65	0.004	0.457	0.108	1.235	63777
MG.A.2	112.27	0.002	0.846	0.237	3.930	236473
MG.A.4	59.84	0.003	0.442	0.128	2.070	123875
MG.A.8	31.38	0.003	0.476	0.114	1.041	60627
MG.B.2	526.28	0.002	0.821	0.238	4.176	236635
MG.B.4	280.11	0.003	0.432	0.130	1.706	123793
MG.B.8	148.29	0.003	0.442	0.116	0.893	60600
LU.A.2	2116.54	0.002	0.110	0.030	0.532	28754
LU.A.4	1102.50	0.002	0.069	0.017	0.255	14915
LU.A.8	574.47	0.003	0.067	0.016	0.192	8655
LU.B.2	9712.87	0.002	0.357	0.104	1.734	101975
LU.B.4	4757.80	0.003	0.190	0.056	0.808	53522
LU.B.8	2444.05	0.004	0.222	0.057	0.548	30134
EP.A.2	123.81	0.002	0.010	0.003	0.074	1834
EP.A.4	61.92	0.003	0.011	0.004	0.073	1743
EP.A.8	31.06	0.004	0.017	0.005	0.073	1661
EP.B.2	495.49	0.001	0.009	0.003	0.196	2011
EP.B.4	247.69	0.002	0.012	0.004	0.122	1663
EP.B.8	126.74	0.003	0.017	0.005	0.083	1656

1.8.3 其它

有的同学不想让某个表格或者图片出现在索引里面，那么请使用命令 `\caption*{}`，这个命令不会给表格编号，也就是出来的只有标题文字而没有“表 XX”，“图 XX”，否则索引里面序号不连续就显得不伦不类，这也是 \LaTeX 里星号命令默认的规则。

有这种需求的多是本科同学的英文资料翻译部分，如果你觉得附录中英文原文中的表格和图片显示成“表”和“图”很不协调的话，一个很好的办法就是用 `\caption*`，参数随便自己写，比如不守规矩的表 1.111 和图 1.111 能满足这种特殊需要（可以参看附录部

分)。

表 1.111 这是一个手动编号，不出现在索引中的表格。

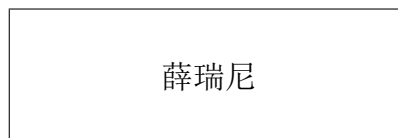


Figure 1.111 这是一个手动编号，不出现在索引中的图。

如果你的确想让它编号，但又不想让它出现在索引中的话，那就自己看看代码改一改吧，我目前不打算给模板增加这种另类命令。

最后，虽然大家不一定会独立使用小页，但是关于小页中的脚注还是有必要提一下。请看下面的例子。

柳宗元，字子厚（773-819），河东（今永济县）人^①，是唐代杰出的文学家，哲学家，同时也是一位政治改革家。与韩愈共同倡导唐代古文运动，并称韩柳^②。

① 山西永济水饺。

② 唐宋八大家之首二位。

唐朝安史之乱后，宦官专权，藩镇割据，土地兼并日渐严重，社会生产破坏严重，民不聊生。柳宗元对这种社会现实极为不满，他积极参加了王叔文领导的“永济革新”，并成为这一运动的中坚人物。他们革除弊政，打击权奸，触犯了宦官和官僚贵族利益，在他们的联合反扑下，改革失败了，柳宗元被贬为永州司马。

1.9 定理环境

给大家演示一下各种和证明有关的环境：

假设 1.1：待月西厢下，迎风户半开；隔墙花影动，疑是玉人来。

$$c = a^2 - b^2 \quad (1-1)$$

$$= (a + b)(a - b) \quad (1-2)$$

千辛万苦，历尽艰难，得有今日。然相从数千里，未曾衰戚。今将渡江，方图百年欢笑，如何反起悲伤？（引自《杜十娘怒沉百宝箱》）

定义 1.1：子曰：「道千乘之国，敬事而信，节用而爱人，使民以时。」

千古第一定义！问世间、情为何物，只教生死相许？天南地北双飞客，老翅几回寒暑。欢乐趣，离别苦，就中更有痴儿女。君应有语，渺万里层云，千山暮雪，只影向谁去？

横汾路，寂寞当年箫鼓，荒烟依旧平楚。招魂楚些何嗟及，山鬼暗啼风雨。天也妒，未信与，莺儿燕子俱黄土。千秋万古，为留待骚人，狂歌痛饮，来访雁丘处。

命题 1.1: 曾子曰：「吾日三省吾身——为人谋而不忠乎？与朋友交而不信乎？传不习乎？」

多么凄美的命题啊！其日牛马嘶，新妇入青庐，奄奄黄昏后，寂寂人定初，我命绝今日，魂去尸长留，揽裙脱丝履，举身赴清池，府吏闻此事，心知长别离，徘徊庭树下，自挂东南枝。

注释 1.1: 天不言自高，水不言自流。

$$\begin{aligned}\varphi(x, z) &= z - \gamma_{10}x - \gamma_{mn}x^m z^n \\ &= z - Mr^{-1}x - Mr^{-(m+n)}x^m z^n\end{aligned}$$

$$\zeta^0 = (\xi^0)^2, \quad (1-3)$$

$$\zeta^1 = \xi^0 \xi^1, \quad (1-4)$$

$$\zeta^2 = (\xi^1)^2, \quad (1-5)$$

天尊地卑，乾坤定矣。卑高以陈，贵贱位矣。动静有常，刚柔断矣。方以类聚，物以群分，吉凶生矣。在天成象，在地成形，变化见矣。鼓之以雷霆，润之以风雨，日月运行，一寒一暑，乾道成男，坤道成女。乾知大始，坤作成物。乾以易知，坤以简能。易则易知，简则易从。易知则有亲，易从则有功。有亲则可久，有功则可大。可久则贤人之德，可大则贤人之业。易简，而天下矣之理矣；天下之理得，而成位乎其中矣。

公理 1.1: 两点间直线段距离最短。

$$x \equiv y + 1 \pmod{m^2} \quad (1-6)$$

$$x \equiv y + 1 \pmod{m^2} \quad (1-7)$$

$$x \equiv y + 1 \pmod{m^2} \quad (1-8)$$

《彖曰》：大哉乾元，万物资始，乃统天。云行雨施，品物流形。大明始终，六位时成，时乘六龙以御天。乾道变化，各正性命，保合大和，乃利贞。首出庶物，万国咸宁。

《象曰》：天行健，君子以自强不息。潜龙勿用，阳在下也。见龙再田，德施普也。终日乾乾，反复道也。或跃在渊，进无咎也。飞龙在天，大人造也。亢龙有悔，盈不可久也。用九，天德不可为首也。

引理 1.1：《猫和老鼠》是我最爱看的动画片。

$$\begin{aligned} \int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2] - 2f(x)g(x)f(y)g(y) dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \end{aligned}$$

行行重行行，与君生别离。相去万余里，各在天一涯。道路阻且长，会面安可知。胡马依北风，越鸟巢南枝。相去日已远，衣带日已缓。浮云蔽白日，游子不顾返。思君令人老，岁月忽已晚。弃捐勿复道，努力加餐饭。

定理 1.1：犯我强汉者，虽远必诛

——陈汤（汉）

$$y = 1 \quad (1-9a)$$

$$y = 0 \quad (1-9b)$$

道可道，非常道。名可名，非常名。无名天地之始；有名万物之母。故常无，欲以观其妙；常有，欲以观其徼。此两者，同出而异名，同谓之玄。玄之又玄，众妙之门。上善若水。水善利万物而不争，处众人之所恶，故几于道。曲则全，枉则直，洼则盈，敝则新，少则多，多则惑。人法地，地法天，天法道，道法自然。知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者寿。

证明 燕赵古称多感慨悲歌之士。董生举进士，连不得志于有司，怀抱利器，郁郁适兹土，吾知其必有合也。董生勉乎哉？

夫以子之不遇时，苟慕义强仁者，皆爱惜焉，矧燕、赵之士出乎其性者哉！然吾尝闻风俗与化移易，吾恶知其今不异于古所云邪？聊以吾子之行卜之也。董生勉乎哉？

吾因子有所感矣。为我吊望诸君之墓，而观于其市，复有昔时屠狗者乎？为我谢曰：“明天子在上，可以出而仕矣！”

——韩愈《送董邵南序》

□

推论 1.1：四川话配音的《猫和老鼠》是世界上最好看最好听最有趣的动画片。

$$V_i = v_i - q_i v_j, \quad X_i = x_i - q_i x_j, \quad U_i = u_i, \quad \text{for } i \neq j; \quad (1-10)$$

$$V_j = v_j, \quad X_j = x_j, \quad U_j u_j + \sum_{i \neq j} q_i u_i. \quad (1-11)$$

迢迢牵牛星，皎皎河汉女。纤纤擢素手，札札弄机杼。终日不成章，泣涕零如雨。河汉清且浅，相去复几许。盈盈一水间，脉脉不得语。

例 1.1：大家来看这个例子。

$$\begin{cases} \nabla f(\mathbf{x}^*) - \sum_{j=1}^p \lambda_j \nabla g_j(\mathbf{x}^*) = 0 \\ \lambda_j g_j(\mathbf{x}^*) = 0, \quad j = 1, 2, \dots, p \\ \lambda_j \geq 0, \quad j = 1, 2, \dots, p. \end{cases} \quad (1-12)$$

练习 1.1：清列出 Andrew S. Tanenbaum 和 W. Richard Stevens 的所有著作。

猜想 1.1: *Poincare Conjecture* If in a closed three-dimensional space, any closed curves can shrink to a point continuously, this space can be deformed to a sphere.

问题 1.1：回答还是不回答，是个问题。

如何引用定理 1.1 呢？加上 `label` 使用 `ref` 即可。妾发初覆额，折花门前剧。郎骑竹马来，绕床弄青梅。同居长干里，两小无嫌猜。十四为君妇，羞颜未尝开。低头向暗壁，千唤不一回。十五始展眉，愿同尘与灰。常存抱柱信，岂上望夫台。十六君远行，瞿塘滟滪堆。五月不可触，猿声天上哀。门前迟行迹，一一生绿苔。苔深不能扫，落叶秋风早。八月蝴蝶来，双飞西园草。感此伤妾心，坐愁红颜老。

1.10 参考文献

当然参考文献可以直接写 `bibitem`，虽然费点功夫，但是好控制，各种格式可以自己随意改写。

本模板推荐使用 `BIBTEX`，样式文件为 `bnubib.bst`，基本符合学校的参考文献格式（如专利等引用未加详细测试）。看看这个例子，关于书的^[28–30]，还有这些^[31–33]，关于杂志的^[34–36]，硕士论文^[37,38]，博士论文^[39,40]，标准文件^[41]，会议论文^[42,43]，技术报告^[44]。中文参考文献^[45]应增加 `lang='zh'` 字段，以便进行相应处理。另外，这个 `bst` 对中文文献^[46]的支持并不是十全十美，如果有不如意的地方，请手动修改 `bbl` 文件。

有时候不想要上标，那么可以这样^[39]，这个非常重要。

1.11 公式

贝叶斯公式如式 (1-13)，其中 $p(y|\mathbf{x})$ 为后验； $p(\mathbf{x})$ 为先验；分母 $p(\mathbf{x})$ 为归一化因子。

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \quad (1-13)$$

论文里面公式越多， \TeX 就越 happy。再看一个 \LaTeX 的例子：

$$\det \mathbf{K}(t = 1, t_1, \dots, t_n) = \sum_{I \in \mathbf{n}} (-1)^{|I|} \prod_{i \in I} t_i \prod_{j \in I} (D_j + \lambda_j t_j) \det \mathbf{A}^{(\lambda)}(\bar{I}|\bar{I}) = 0. \quad (1-14)$$

前面定理示例部分列举了很多公式环境，可以说把常见的情况都覆盖了，大家在写公式的时候一定要好好看 \LaTeX 的文档，并参考模板中的用法：

$$\begin{aligned} \int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2] - 2f(x)g(x)f(y)g(y) dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \end{aligned}$$

其实还可以看看这个多级规划：

$$\left\{ \begin{array}{l} \max_x F(x, y_1^*, y_2^*, \dots, y_m^*) \\ \text{subject to:} \\ G(x) \leq 0 \\ (y_1^*, y_2^*, \dots, y_m^*) \text{ solves problems } (i = 1, 2, \dots, m) \\ \left\{ \begin{array}{l} \max_{y_i} f_i(x, y_1, y_2, \dots, y_m) \\ \text{subject to:} \\ g_i(x, y_1, y_2, \dots, y_m) \leq 0. \end{array} \right. \end{array} \right. \quad (1-15)$$

这些跟规划相关的公式都来自于刘宝碇老师《不确定规划》的课件。

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[47]与[48]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[4,48,49]等都广泛采用了等效RC电路方法进行热分析建模，并采用如下公式进行工作温度的求解式中 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：

式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

由于与工作电压成正比，我们可以得到动态功耗的计算公式

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

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下式计算

其中，

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[49]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[48]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1所示。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给出了能耗的数据，图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℃），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℃），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℃），最高可超过380K（107℃），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。

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， $w_t = (w_{1t}, w_{2t}, \dots, w_{Nt})$ 对其中任一权重分量 w_{kt} 可以采用下式进行计算

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，则开始进行决策：

式中 $DPTM_t$ 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[50]。

2.7 其它例子

在第 1 章中我们学习了贝叶斯公式 (1-13)，这里我们复习一下：

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \quad (2-1)$$

2.7.1 绘图

本模板不再预先装载任何绘图包（如 `pstricks`，`pgf` 等），完全由你自己来决定。个人觉得 `pgf` 不错，不依赖于 `Postscript`。此外还有很多针对 \LaTeX 的 GUI 作图工具，如 `XFig(jFig)`, `WinFig`, `Tpx`, `Ipe`, `Dia`, `Inkscape`, `LaTeXPiX`, `jPicEdt`, `jaxdraw` 等等。

2.7.2 插图

强烈推荐《 $\text{\LaTeX} 2_{\epsilon}$ 插图指南》！关于子图形的使用细节请参看 `subfig` 的说明文档。

2.7.2.1 一个图形

一般图形都是处在浮动环境中。之所以称为浮动是指最终排版效果图形的位置不一定与源文件中的位置对应^①，这也是刚使用 \LaTeX 同学可能遇到的问题。如果要强制固定浮动图形的位置，请使用 `float` 宏包，它提供了 `[H]` 参数。比如图 2.1。

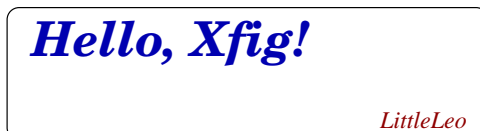


图 2.1 利用 Xfig 制图

大学之道，在明明德，在亲民，在止于至善。知止而后有定；定而后能静；静而后能安；安而后能虑；虑而后能得。物有本末，事有终始。知所先后，则近道矣。古之欲明明德于天下者，先治其国；欲治其国者，先齐其家；欲齐其家者，先修其身；欲修其身者，先正其心；欲正其心者，先诚其意；欲诚其意者，先致其知；致知在格物。物格而后知至；知至而后意诚；意诚而后心正；心正而后身修；身修而后家齐；家齐而后国治；国治而后天下平。自天子以至于庶人，壹是皆以修身为本。其本乱而未治者否矣。其所厚者薄，而其所薄者厚，未之有也！

——《大学》

^① This is not a bug, but a feature of \LaTeX !

古之学者必有师。师者，所以传道受业解惑也。人非生而知之者，孰能无惑？惑而不从师，其为惑也，终不解矣。生乎吾前，其闻道也固先乎吾，吾从而师之；生乎吾後，其闻道也亦先乎吾，吾从而师之。吾师道也，夫庸知其年之先後生於吾乎！是故无贵无贱无长无少，道之所存，师之所存也。

嗟乎！师道之不传也久矣，欲人之无惑也难矣。古之圣人，其出人也远矣，犹且从师而问焉；今之众人，其下圣人也亦远矣，而耻学於师。是故圣益圣，愚益愚。圣人之所以为圣，愚人之所以为愚，其皆出於此乎？爱其子，择师而教之，於其身也，则耻师焉，惑焉。彼童子之师，授之书而习其句读者，非吾所谓传其道、解其惑者也。句读之不知，惑之不解，或师焉，或不焉，小学而大遗，吾未见其明也。巫医、乐师、百工之人不耻相师，士大夫之族曰“师”曰“弟子”之云者，则群聚而笑之。问之，则曰：彼与彼年相若也，道相似也，位卑则足羞，官盛则近谀。呜呼！师道之不复，可知矣。巫医、乐师、百工之人。吾子不齿，今其智乃反不能及，其可怪也欤！圣人无常师。孔子师郯子、苌子、师襄、老聃。郯子之徒，其贤不及孔子。孔子曰：“三人行，必有我师。”是故弟子不必不如师，师不必贤於弟子。闻道有先後，术业有专攻，如是而已。

李氏子蟠，年十七，好古文、六艺，经传皆通习之，不拘於时，学於余。余嘉其能行古道，作师说以贻之。

——韩愈（唐）

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 Floorplan-
ning: 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 Confer-
ence. 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: Mod-
eling 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 Appli-
cations[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 Optimiza-
tion 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. Donald E. Knuth. The \TeX Book[M]. 15th ed., Reading, MA: Addison-Wesley Publishing Company, 1989.
29. Michel Goossens, Frank Mittelbach, Alexander Samarin. The \LaTeX Companion[M]. Reading, MA: Addison-Wesley Publishing Company, 1994: 112–125.
30. P. Gröning, L. Nilsson, P. Ruffieux, et al. Encyclopedia of Nanoscience and Nanotechnology, volume 1. American Scientific Publishers, 2004: 547–579.
31. N. Krasnogor. Towards robust memetic algorithms[C]. In: W.E. Hart, N. Krasnogor, J.E. Smith, (eds.). Proceedings of Recent Advances in Memetic Algorithms. New York: Springer Berlin Heidelberg, 2004: 185–207.
32. 阎真. 沧浪之水. 人民文学出版社, 2001: 185–207.
33. 班固. 苏武传[C]. 见: 郑在瀛, 汪超宏, 周文复, 编. 传记散文英华. 武汉: 湖北人民出版社, 1998: 65–69.
34. M. Chafik El Idrissi, A. Roney, C. Frigon, et al. Measurements of total kinetic-energy released to the $N = 2$ dissociation limit of H_2 — evidence of the dissociation of very high vibrational Rydberg states of H_2 by doubly-excited states[J]. Chemical Physics Letters, 1994, 224(10):260–266.
35. A. Mellinger, C. R. Vidal, Ch. Jungen. Laser reduced fluorescence study of the carbon-monoxide nd triplet Rydberg series-experimental results and multichannel quantum-defect analysis[J]. J. Chem. Phys., 1996, 104(5):8913–8921.
36. Michael Shell. How to Use the IEEEtran \LaTeX Class[J]. Journal of \LaTeX Class Files, 2002, 12(4):100–120.
37. 猪八戒. 论流体食物的持久保存[D]. 北京: 广寒宫大学, 2005.
38. Ashwin Raju Jeyakumar. Metamori: A library for Incremental File Checkpointing[D]. Blacksburg: Virginia Tech, June 21, 2004.

39. 沙和尚. 北京: 清华大学, 2005.
40. Erez Zadok. USA: Computer Science Department, Columbia University, May, 2001.
41. IEEE Std 1363-2000. IEEE Standard Specifications for Public-Key Cryptography[M]. New York: IEEE, 2000.
42. Sangbum Kim, Namyoon Woo, Heon Y. Yeom, et al. Design and Implementation of Dynamic Process Management for Grid-enabled MPICH[C]. Proceedings of the 10th European PVM/MPI Users' Group Conference, Venice, Italy, 2003.
43. C. Kocher, J. Jaffe, B. Jun. Differential Power Analysis[C]. In: M. Wiener, (eds.). Proceedings of Advances in Cryptology (CRYPTO '99), volume 1666 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. 388–397.
44. Alex Woo, David Bailey, Maurice Yarrow, et al. The NAS Parallel Benchmarks 2.0[R]. Technical report, The Pennsylvania State University CiteSeer Archives, December 05, 1995. <http://www.nasa.org/>.
45. 贾宝玉, 林黛玉, 薛宝钗, 等. 论刘姥姥食量大如牛之现实意义[J]. 红楼梦杂谈, 1800, 224:260–266.
46. 王重阳, 黄药师, 欧阳峰, 等. 武林高手从入门到精通[C]. 第 N 次华山论剑, 西安, 中国, 2006.
47. 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.
48. V. Chaturvedi, H. Huang, G. Quan. [C].
49. 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.
50. 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.

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

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

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	600	2%	10%	50%	Max. Connectivity
	Transit-Stub					
Inet-2.1		6000				
Xue	Rui	Ni	BNUT _{HESIS}			
	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 \mathcal{R}^n . Whether derivatives are available or not, the usual strategy is first to select a point in \mathcal{R}^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	600	2%	10%	50%	Max. Connectivity
	Transit-Stub					
Inet-2.1		6000				
Xue	Rui	Ni	BNUT _{HESIS}			
	ABCDEF					

表 附 2.1 正常附录表格的例子

Network Topology		# of nodes	# of clients			Server
GT-ITM	Waxman	600	2%	10%	50%	Max. Connectivity
	Transit-Stub					
Inet-2.1		6000				
Xue	Rui	Ni	BNUT _{HESIS}			
	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，而况大瓠乎！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，謦然已解，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，柳下季之弟名曰盗跖。盗跖从卒九千人，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，弟为盗跖，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，往见盗跖。

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 xxx 教授和物理系 xxx 副教授对本人的精心指导。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