

信息科学技术学术著作丛书

面向嵌入式系统的绿色编译 理论与方法研究

何炎祥 陈 勇 著



科学出版社

信息科学技术学术著作丛书

面向嵌入式系统的绿色编译 理论与方法研究

何炎祥 陈 勇 著

科学出版社

北 京

内 容 简 介

编译器作为软件开发和构建的重要工具,在对软件进行系统化、结构化的优化方面具有无可比拟的优势。随着嵌入式系统的广泛普及和应用,其绿色指标已受到人们的广泛关注。而嵌入式系统是软硬件的结合体,其绿色指标不但受硬件工艺和技术的制约,更受到其上运行软件的影响。本书从能耗以及资源的均衡使用度这两个影响嵌入式系统绿色指标的问题出发,系统而全面地介绍绿色编译优化理论及其关键技术嵌入式系统中的应用。

本书大部分内容是作者近年来的最新研究成果,具有较强的原创性,可作为高等院校和科研院所计算机科学与技术、软件工程、计算机应用技术等相关学科高年级本科生或研究生的教材,也可供软件优化等相关领域的研究人员借鉴。

图书在版编目(CIP)数据

面向嵌入式系统的绿色编译理论与方法研究/何炎祥,陈勇著. —北京:科学出版社, 2014

(信息科学技术学术著作丛书)

ISBN 978-7-03-040501-2

I. 面… II. ①何… ②陈… III. 编译器 IV. TP314

中国版本图书馆 CIP 数据核字(2014)第 082737 号

责任编辑:魏英杰 / 责任校对:李 影

责任印制:张 倩 / 封面设计:陈 敬

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

新科印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2014 年 5 月第 一 版 开本:720×1000 1/16

2014 年 5 月第一次印刷 印张:12 1/2

字数:252 000

定价:65.00 元

(如有印装质量问题,我社负责调换)

《信息科学技术学术著作丛书》序

21 世纪是信息科学技术发生深刻变革的时代,一场以网络科学、高性能计算和仿真、智能科学、计算思维为特征的信息科学革命正在兴起。信息科学技术正在逐步融入各个应用领域并与生物、纳米、认知等交织在一起,悄然改变着我们的生活方式。信息科学技术已经成为人类社会进步过程中发展最快、交叉渗透性最强、应用面最广的关键技术。

如何进一步推动我国信息科学技术的研究与发展;如何将信息技术发展的新理论、新方法与研究成果转化为社会发展的新动力;如何抓住信息技术深刻发展变革的机遇,提升我国自主创新和可持续发展的能力?这些问题的解答都离不开我国科技工作者和工程技术人员的求索和艰辛付出。为这些科技工作者和工程技术人员提供一个良好的出版环境和平台,将这些科技成就迅速转化为智力成果,将对我国信息科学技术的发展起到重要的推动作用。

《信息科学技术学术著作丛书》是科学出版社在广泛征求专家意见的基础上,经过长期考察、反复论证之后组织出版的。这套丛书旨在传播网络科学和未来网络技术,微电子、光电子和量子信息技术、超级计算机、软件和信息存储技术,数据知识化和基于知识处理的未来信息服务业,低成本信息化和用信息技术提升传统产业,智能与认知科学、生物信息学、社会信息学等前沿交叉科学,信息科学基础理论,信息安全等几个未来信息科学技术重点发展领域的优秀科研成果。丛书力争起点高、内容新、导向性强,具有一定的原创性;体现出科学出版社“高层次、高质量、高水平”的特色和“严肃、严密、严格”的优良作风。

希望这套丛书的出版,能为我国信息科学技术的发展、创新和突破带来一些启迪和帮助。同时,欢迎广大读者提出好的建议,以促进和完善丛书的出版工作。

中国工程院院士

原中国科学院计算技术研究所所长



前 言

地球是人类赖以生存的家园,保护地球生态环境的和谐发展是我们应尽的责任和义务。随着现代科技朝着自动化、智能化的快速发展,温室效应、电子垃圾等现代科技的产物却给生态环境带来了难以磨灭的印记,人们在享受现代科技带来便利的同时也不可避免地受其影响,如何使现代科技在丰富人类生产生活的同时兼顾环境的和谐发展已经成为不容忽视的重要问题。

计算机作为现代科技的重要产物,其应用已经涉及人们日常生活的方方面面。特别是随着电子芯片的日益小型化、智能化,各类嵌入式产品已经得到广泛应用,其消耗的能源和产生的电子垃圾等已经成为生态破坏的重要因素之一。解决这些嵌入式电子设备带来的负面影响是解决生态环境和谐发展的必经之路,许多工业界和学术界的研究者都对其给予了高度关注。

然而,嵌入式系统是软硬件的结合体,要降低能耗,减缓电子垃圾产生的速度,一方面需要对硬件设备、制作工艺不断更新,使其满足低能耗、高耐久性的要求;另一方面更是需要对其运行的软件进行优化,使其能够合理地利用各种硬件资源,充分发挥各硬件资源的自身优势。编译器作为软件开发和构建的重要工具,一直以来都是对软件进行系统化和结构化优化的主力军。因此,从软件方面解决嵌入式设备能耗和电子垃圾的问题,编译器将拥有得天独厚的优势。

绿色本是大自然中一种很普通的颜色,大部分的植物具有该种颜色。但由于人们看到绿色时总有种生机勃勃、回归大自然的感觉,绿色也象征着大自然的和谐美好,因而不少与生态环保相关的事物会冠以该词,如绿色蔬菜、绿色贸易、绿色食品等。在计算机领域也是如此,不少以降低系统能耗、减少电子垃圾排放的技术也是以绿色为主导,如绿色计算、绿色网络、绿色操作系统等。因此,本书将所有与降低系统能耗、提高设备资源有效利用率,减少电子垃圾排放的编译技术称为绿色编译技术。

为缓解嵌入式系统广泛应用对生态环境的负面影响,促进各类智能嵌入式设备和谐健康的快速发展,本书将以嵌入式系统中与生态环境相关的两个主要问题——系统运行的能耗和系统资源使用的均衡度——为着眼点,从编译器的角度对其关键技术和优化方法进行系统而深入的阐述。

本书共 8 章,各章的主要内容组织如下。

第 1 章主要介绍绿色编译的研究背景及国内外研究的现状,以便读者初步认识现有的绿色编译技术。

第2章根据现有绿色相关概念,给出绿色编译器的定义。在此基础上,根据其 主要特征介绍绿色编译优化框架和绿色评估模型,以有效地指导和评估绿色编译 优化效果。

第3章主要介绍两种低能耗体系结构下的绿色指令调度方法。一种主要面向 TS 处理器,结合空操作指令的填充,利用图博弈模型对指令序列进行调度,以减少 因数据依赖而导致的数据前送操作,提高 TS 处理器的能效和系统的绿色指标。 另一种主要面向总线翻转编码体系结构,分析绿色评估模型中总线相关的绿色评 估指标,并以此为标准,给出一种反馈信息指导的绿色指令调度算法。同时,利用 程序执行的动态反馈信息,结合总线翻转编码的特点,充分减少相邻总线之间的翻 转次数,均衡各总线之间的翻转负载,以提高系统的整体绿色指标。

第4章主要以通用存储系统中寄存器以及栈数据分配为出发点,以提高存储 系统和总线系统的绿色指标为主要目标,介绍一种多目标数据分配优化方法:可交 换类指令操作数重排优化,基于扩展图着色的寄存器重分配优化和面向栈数据均 衡访问的数据重分配优化。通过这些优化手段,对程序中访问的寄存器和栈数据 进行调整,以获得较均衡的寄存器访问频度和栈存储单元访问频度,减少指令数据 总线的动态翻转能耗,提高系统整体的绿色指标。

第5章根据新型存储技术的特征,针对混合便签式存储器、混合缓存以及易失 性 STT-RAM 缓存,介绍以整数线性规划算法和启发式算法为主要内容的栈数据 分配方法,以最大程度降低存储系统的能耗,提高系统的绿色指标。

第6章介绍基于符号执行的能耗错误检测及反例生成技术。首先通过对能耗 错误特征的总结和分析,给出一种基于符号执行的能耗错误检测方法。该方法利 用程序的过程内分析,获得单个函数的符号执行信息,然后根据过程间分析对单个 函数的符号执行信息进行全局综合,进而获得较为准确的能耗错误相关信息。同 时,符号执行记录对应的分支路径信息,利用该信息不但能够较好地生成对应的测 试用例,而且可以结合约束求解器快速定位错误位置,为开发出高绿色指标的软件 提供保障。

第7章结合现有的一些新的理论和算法,对可能的新型绿色编译优化方法进 行探索,期望为后续的绿色编译研究提供一个可行的研究思路。

第8章总结全书并展望后续研究。

本书是武汉大学计算机学院众多科研人员多年学习、研究和工程实践沉淀的 成果。参与相关研究的人员包括刘陶、吴伟、陈勇、李清安、刘健博、徐超、胡明昊、 董伟、廖希密、刘钱、陈念、吴昊、毋国庆、文卫东、吴黎兵、彭敏、李飞、邵凌霜、贾向 阳、严飞、余发江等,其中第5章主要由李清安参与撰写,其余章节主要由陈勇参与 撰写,何炎祥具体规划和设计了全书的内容并进行了统稿。文卫东、吴黎兵、彭敏 对本书的初稿提出了很多建设性意见,在此对他们的积极参与和热心帮助表示衷

心的感谢。

本书是国内第一本针对绿色编译展开系统研究的著作,对相关领域的研究人员具有一定的借鉴意义和参考价值。本书的出版得到国家自然科学基金“可信软件基础研究”重大研究计划项目“可信编译理论与实现方法研究”(2009,项目编号:90818018)和“基于编译的高可信嵌入式软件开发与验证方法研究”(2011,项目编号:91018009)、国家自然科学基金“基于编译的嵌入式系统优化研究”(2012,项目编号:61170022),国家自然科学基金“可信软件基础研究”重大研究计划重点项目“可信软件构造理论与方法研究”(2012,项目编号:9111800)、国家自然科学基金“面向嵌入式系统绿色需求的编译理论与方法研究”(2014,项目编号:61373039),以及湖北省自然科学基金“可信计算的软件理论与关键技术研究(2009,项目编号:2008CDA007)”等项目的资助,在此一并表示感谢。

绿色编译理论及其应用是当前处于科学前沿的研究课题之一,相关的理论和技术还在发展之中,许多新的思想、理论和方法还需要进一步验证和完善,由于作者的水平和经验有限,书中不妥之处在所难免,恳请读者给予批评指正,共同推进绿色编译理论研究的进步和发展。



2014年1月于武汉

目 录

《信息科学技术学术著作丛书》序

前言

第 1 章 绪论	1
1.1 研究背景	1
1.1.1 绿色需求对现代嵌入式系统的重要性	1
1.1.2 编译器对嵌入式软件绿色优化的重要性	2
1.2 国内外研究现状	3
1.2.1 面向总线系统的绿色优化技术研究	3
1.2.2 面向存储系统的绿色优化技术研究	8
1.2.3 面向处理器的绿色优化技术研究	13
1.2.4 软件自动化测试及能耗调试相关研究	16
1.3 组织结构	18
1.4 本章小结	18
参考文献	18
第 2 章 绿色编译及评估模型	26
2.1 绿色编译器定义	26
2.2 绿色编译优化框架	27
2.3 绿色评估模型	31
2.3.1 绿色编译优化评估模型	31
2.3.2 能耗错误检测评估模型	32
2.4 本章小结	33
参考文献	33
第 3 章 面向嵌入式系统的指令调度方法	35
3.1 面向时序推测处理器绿色指标的指令调度方法	35
3.1.1 时序推测处理器简介	35
3.1.2 图博弈模型	38
3.1.3 编译器可控的时序推测处理器绿色评估指标	39
3.1.4 基于图博弈模型的指令调度方法	41
3.1.5 实验与结果分析	55
3.2 面向总线翻转编码的绿色指令调度方法	59

3.2.1 总线绿色评估模型	59
3.2.2 总线翻转编码	61
3.2.3 反馈信息指导的面向总线的绿色指令调度算法	61
3.2.4 实验结果与分析	65
3.3 本章小结	68
参考文献	69
第4章 多目标数据分配优化方法	71
4.1 数据分配对系统绿色指标的影响	71
4.2 绿色评估模型指导的多目标数据分配优化方法	73
4.2.1 多目标数据分配总体优化框架	73
4.2.2 可交换类指令操作数重排优化	74
4.2.3 面向绿色需求的寄存器重分配方法	76
4.2.4 面向存储系统绿色指标的栈数据分配方法	85
4.2.5 实验结果与分析	89
4.3 本章小结	93
参考文献	93
第5章 面向新型存储技术的绿色编译优化方法	96
5.1 新型存储体系结构对绿色优化的影响	96
5.1.1 PCM 面临的挑战及解决方案	99
5.1.2 STT-RAM 面临的挑战及解决方案	100
5.2 面向混合便签式存储器的低能耗数据分配方法	100
5.2.1 背景知识	101
5.2.2 数据分配整数线性规划方法	103
5.2.3 迭代图着色算法	105
5.2.4 实验结果与分析	111
5.3 面向易失性 STT-RAM 缓存的低功耗编译优化方法	114
5.3.1 缓存刷新机制简介	115
5.3.2 数据分配方案同刷新频度的关系	117
5.3.3 数据分配整数线性规划解决方案	119
5.3.4 启发式分配方法	122
5.3.5 实验结果与分析	125
5.4 面向混合缓存的低功耗编译技术	136
5.4.1 混合缓存简介	137
5.4.2 混合缓存中迁移操作导致的负载	138
5.4.3 基于缓存加锁的方法	142

5.4.4 实验结果与分析	145
5.5 本章小结	150
参考文献	150
第6章 基于符号执行的能耗错误检测及反例生成技术研究	154
6.1 能耗错误简介	154
6.2 符号执行技术	155
6.3 基于符号执行技术的能耗错误分析方法	157
6.3.1 能耗错误过程内分析	157
6.3.2 能耗错误过程间分析	164
6.4 应用举例	165
6.5 实验评估	171
6.5.1 实验构建	171
6.5.2 结果分析	172
6.6 本章小结	172
参考文献	173
第7章 绿色编译新型优化方法探究	175
7.1 基于计算博弈论的分块多层次绿色编译优化	175
7.2 基于总线翻转编码和多维度集成学习的绿色编译技术	177
7.3 基于编译的多核环境下的并行程序绿色优化	178
7.3.1 基于数据依赖的程序数据级并行性挖掘	178
7.3.2 基于编译的异构多核环境下的低能耗多因素优化	180
7.4 基于编译的动静态结合能耗错误检测和定位方法	182
7.5 本章小结	183
参考文献	183
第8章 总结与展望	185
后记——三言两语	188

第1章 绪 论

1.1 研究背景

1.1.1 绿色需求对现代嵌入式系统的重要性

在信息化、电子化高速发展的今天,各种便携式移动设备,如智能手机、平板电脑等被广泛应用于人们的日常生活和工作当中,极大地方便和丰富了人们的物质、文化生活。与此同时,电子产品的快速普及不但消耗了巨大的能源(仅个人计算机消耗的能源就占到全部电力的12%^[1]),释放了大量的温室气体,而且废弃的电子垃圾中含有铅、铬、镉和汞等有毒物质,任意排放将严重污染土壤、水等环境资源,给人们的正常生活带来不可忽视的负面影响^[2]。全球权威技术调查机构高德纳(Gartner)对IT业碳排放量的统计结果显示,IT业的二氧化碳排放量已经占全球碳排放总量的2%,比重等同于航空业^[3]。而且巨大的能源消耗大大增加了IT产品本身的成本,2010年全球IT技术的花费已经达到3万亿美元。McKinsey报告指出,到2020年,手提电脑、智能手机、平板电脑等各种IT设备将成为温室气体排放量最大的源头之一^[4]。在电子垃圾方面,联合国环境规划署(UNEP)^[5]相关资料指出,2010年美国产生了300万吨电子垃圾,中国产生了230万吨的电子垃圾,全球每年将产生4000万吨的电子垃圾。据其预测,如果仍不采取相应保护手段,到2020年,各个国家在各个领域产生的电子垃圾将是现在的2~18倍。

进入21世纪以来,随着半导体技术的进步,移动智能终端设备得到了迅猛发展,其产品种类越来越多,功能越来越丰富,应用范围也越来越广泛,涉及人们工作、学习和生活的方方面面,同时人们对其使用的频度和数量也稳步增长。日本市场调研机构富士总研(Fuji Chimera Research)发表的研究报告称,2012年全球手机市场规模为15.9亿部,其中智能手机为7.4亿部,占46.6%。报告预计,2017年全球手机销量将达19.7亿部,与2012年相比增长24.1%;其中智能手机将达15.05亿部,增长103%^[6]。就电子垃圾而言,由于手机、平板电脑、车载导航仪等便携式移动嵌入式设备数量多、损耗快,因此其造成的电子垃圾不但数量大,而且增速快。据高德纳预计,2020年我国仅废弃手机产生的电子垃圾将为2007年的7倍左右,而印度将达到2007年的18倍。2013年绿色计算所带来的市场份额有可能达到了48亿美元^[7]。

1.1.2 编译器对嵌入式软件绿色优化的重要性

针对上述严峻的能源环境问题,以低能耗、低污染、低成本和高功效为主要目标的绿色技术已经成为人们关注的焦点。从 2008~2010 年高德纳公布的技术发展趋势报告可以看出,绿色 IT 技术已经成为了十大 IT 关键技术之首^[7]。在硬件设备和体系结构的设计上,各主要 IT 公司及计算机供应商均相应的采取了各种措施,以提高产品的绿色指标,减少对环境的污染。戴尔公司致力于其提出的零碳计划,以求最大化能量利用的效率。苹果公司将减少其新产品中有毒重金属的使用,同时增加对旧产品的回收,以减少电子垃圾及其对环境的影响^[1]。三星、诺基亚、索尼等主要移动嵌入式设备制造商^[8~10]均建立了自己的绿色产品研究实验室,研究新型产品,降低对环境的影响。然而,随着各种 IT 设备自动化智能化程度的不断提升,设备的能耗、功效不仅受具体的硬件设备材质的影响,而且也同其上运行的软件息息相关,相同功能的硬件模块采用不同的软件实现方式将产生不同的能耗以及资源的损耗。因此,如何从软件方面提高电子设备的绿色指标也是一个不容忽视的重要问题。

早期的嵌入式软件由于功能比较单一,逻辑相对简单,很多熟练的程序员能够凭借经验,使用较为低级的汇编语言进行编程获得较高质量的可执行程序。随着嵌入式处理器芯片、存储芯片等关键设备制作工艺和性能的飞速发展,嵌入式系统得到广泛普及,其功能越来越全面,运行的应用程序也越来越复杂,但其开发和调试的难度也随之增大。传统的用低级语言进行编程的方式已经很难满足需求,目前的大部分嵌入式程序均采用高级语言,如 C、Java 等进行编写,然后再利用交叉编译器将该程序转换为对应平台的可执行目标程序。因此,编译器作为软件开发和语言转换的工具,不但能够结合具体的嵌入式系统硬件设备信息进行针对性的绿色优化,指导最终的目标程序朝着绿色需求的方向生成,而且能够利用其在程序转换中对源程序的分析结果,提高错误检测的速度和定位精度,帮助程序员尽早发现和修正程序错误,减少软件开发过程中资源和能源的消耗。

然而,传统的编译优化技术主要是针对通用机设计的,利用这类编译器改良的嵌入式交叉编译器虽然能够生成性能较好的嵌入式程序,但相比于直接用低级语言编写的程序,在现有的绿色需求方面仍然有较大的优化空间。究其原因,主要有三个方面。首先,嵌入式系统由于受到体积、功耗等因素的影响,相对于通用机在体系结构上有很大的区别,其可用资源也少了很多,因而导致某些通用机上可以实现的优化在嵌入式系统中难以进行。其次,传统编译优化技术为简化编译器的实现,把代码生成分为指令选择、寄存器分配、指令调度三个独立的阶段。这三个阶段其实是彼此关联的,某个阶段的优化结果可能会影响其他阶段的优化效果。最后,由于传统编译优化技术主要以性能优化为主,较少的考虑能耗、资源等绿色相

关因素的影响。性能优化与绿色优化并非完全一致,一般的性能优化是以资源和能源的消耗(如使用较多的寄存器资源,提高处理器运行速率等)为代价的,难以满足资源和体积受限的嵌入式系统的需求,更难以满足绿色计算的需求。因此,我们在后续章节将详细阐述几种面向嵌入式系统绿色需求的编译优化技术,以弥补现有编译优化技术的不足。

同时,绿色需求的不断增长也给编译调试带来了新的挑战。首先,随着人们对低能耗等绿色需求的不断增长,一类新的错误——能耗错误(ebug)已经逐步受到人们的重视。该类错误不同于传统的功能性错误,它不会引起程序运行结果出错,但会导致大量程序功能无关的能源消耗,而这些功能无关的能源消耗正是绿色需求要解决的问题。这种情况下程序的运行结果是正确的,因此传统以检测程序功能性错误为主要目标的编译调试信息已经无能为力,难以满足绿色编译的需求。其次,随着编程语言朝着自动化、智能化方法的不断改进,编程语言学习的难度越来越低,越来越多的业余人员参与到嵌入式程序特别是各种智能手机应用程序,如Android和iOS平台App的开发中来,而传统的编译调试需要大量的人机交互,程序员首先需要根据出错的测试用例估计出错的大概位置,然后通过断点、变量监视等编译器提供的调试接口逐步跟踪程序运行,以找出程序中的错误。但这些工作对于业余开发人员将是十分困难的工作,他们将花费十倍甚至更多的时间才能完成一个相同错误的检测,而更多的时间花费则意味着更多的资源与能源的消耗。因此,如果能根据能耗错误的特点,提供更为丰富和友好的编译调试和警告信息,对减少开发人员错误查找的时间,提高能耗错误发现的精度和速度,减少软件开发过程中资源和能源的消耗,提高开发过程的绿色程度将是十分必要的。我们将在第6章对这类绿色相关的能耗错误检测和定位方法进行详细讨论。

1.2 国内外研究现状

为从软件方面提高计算机系统能源和资源使用的绿色指标,众多学者从组成计算机系统的三个主要部分,即总线、存储器以及处理器均进行了一定的研究,分别从各自的角度针对不同的体系结构对如何提高计算机系统的绿色指标进行分析和尝试,提出了一些新的面向绿色需求的优化及相关错误检测方法。下面我们分别从总线系统、存储系统、处理器和软件自动化测试及能耗调试这四个方面对相应的研究现状进行分析和介绍。

1.2.1 面向总线系统的绿色优化技术研究

总线是电子设备各模块内部以及各模块之间信号和数据传输的通道,其工作性能直接影响到整个系统的效率及能耗,据文献[11]所述,芯片内部总线的动态能

耗约占芯片总能耗的 70%。随着半导体工艺逐步进入深亚微米甚至是纳米时代,芯片的体积越来越小,总线布局也越来越密集,其能耗所占比例将越来越大,因此总线的能耗、工作的稳定性及可靠性对整个系统的影响愈发重要,要实现绿色计算机系统,有必要对总线传输进行对应的优化处理。

根据总线上传输的数据类型,总线可以分为数据总线、地址总线和控制总线。数据总线主要传输的是各种通信数据,包括各种存储器(如指令存储器、数据存储器等)存储的数据。地址总线主要传输地址数据,包括指令地址以及数据地址等。控制总线主要传输的是控制信息,以控制相关部件的工作状态。无论哪种类型的总线,其能耗及工作的可靠性和稳定性主要受两方面因素的影响,即单根总线本身的影响以及并行传输的相邻总线之间(总线串扰)的影响。

1. 单根总线本身的优化

针对于单根总线本身,主要以 CMOS 电路的动态能耗模型式(1.1)为基础^[12,13],即

$$E_{\text{dyn}} = \alpha C f V_{\text{dd}}^2 t \quad (1.1)$$

其中, α 是翻转因子,表示该总线由 0 变为 1 或者由 1 变为 0 的频率; C 是负载电容; f 和 V_{dd} 分别是工作频率和工作电压; t 是工作时间。

由式(1.1)可以看出,在电压、频率和电容相同的情况下, α 越小,总线的动态能耗也将越小,因此大部分针对单根总线本身的优化主要是根据不同总线类型传输数据的特点,通过减少总线翻转次数来达到减少总线能耗的目的。

减少总线翻转次数的主要方法之一是编码技术。例如,Stan 和 Burleson^[14] 提出翻转编码,Lv 等^[15] 提出基于字典查找的总线编码,Suresh 等^[16] 提出 VALVE 和 TUBE 编码,Benini 等^[17] 提出固定步长增长的 T0 编码,Mehta 等^[18] 及 Guo 等^[19] 提出相邻数据海明距离最短的格雷编码。总线编码的基本思想是待传输的数据在传输前通过编码电路进行编码,使相邻传输的数据引起的总线翻转次数尽可能少。然后在传输接收端利用解码模块对编码后的数据进行解码,以保证传输数据的正确性。

由于采用硬件编码的方法或者需要增加额外的编码以及解码电路,或者需要采用定制模块结构,这些增加的硬件设备不但需要额外的能耗开销,而且相对于越来越小的芯片尺寸,其体积开销也将影响到整个系统的设计难度。此外,定制模块结构,如文献^[19]中采用格雷编码的内存地址将影响该设备的适用范围。总线数据的传输与通信,主要是由其上运行的软件控制的,不少研究者开始从软件方面考虑如何减少总线翻转次数的问题。编译器作为软件生成的主要手段也成为他们选择的主要工具。

目前基于编译技术减少总线翻转次数的方法主要是通过指令调度来减少相邻

指令访问时代码段数据总线之间的翻转次数。Parikh 等^[20]总结了各种低能耗指令调度算法,并通过与性能优先指令调度算法进行的对比,指出性能最佳的指令调度序列并不一定是能耗最低的调度序列。LEE 等^[21]针对 VLIW 体系结构,提出水平调度和垂直调度两种调度方法来减少总线翻转次数,达到降低总线能耗的目的。Shao 等^[22]证明低能耗指令调度问题是 NP 完全问题,并在 VLIW 体系结构中针对总线翻转次数和调度长度,提出三种启发式调度算法,以尽可能获得较低的功耗。最近,Chabini 和 Wolf 针对汇编级基本块内的指令调度,将最小化基本块内总线翻转次数的指令调度问题转化为一个整数线性规划(integer linear programming, ILP)问题。同时,由于该问题是一个 NP 完全问题,使用整数线性规划求解需要大量的时间开销,因此 Chabini 和 Wolf^[23]又设计了两种启发式调度算法,以求快速获得总线翻转次数较少的调度方案。

然而,这些基于编译减少总线翻转的方法主要是单纯从软件的角度出发,并未考虑底层的硬件结构。虽然硬件编码需要额外的电路,但有些编码和解码电路十分简单,开销很低,如 0-1 翻转编码,在体系结构中实现并不困难。因此,我们可以结合这些已经实现的编码技术,在编译优化的过程中采取适应该编码方案的优化方法,将有可能获得更好的优化效果。

除直接的动态能耗外,随着芯片体积的缩小,电子工艺已经进入深亚微米阶段,总线之间的布局越来越密集,因泄露电流而产生的能耗所占比例也越来越大,使每根总线本身的负载也将对系统总的能耗产生重要影响。如果某根总线的翻转频率远大于其他总线,其动态能耗会导致该总线温度远大于其他总线。温度的增加往往会增加总线的泄漏能耗,同时也会导致其可靠性下降。错误的传输要么需要纠错电路,要么需要重传数据,这也将间接增加系统总的能耗。目前基于编译的低能耗指令调度主要以减少总线总的翻转次数为主,即最小化式(1.2),但对单根总线的峰值翻转次数考虑较少。我们以 Mibench 作为基准测试用例集,以 ARM 为目标指令集,对其执行过程中用于传输代码段的每根指令数据总线的动态负载(0-1 翻转次数)进行了统计,其结果如图 1.1 所示。可以看出,各总线负载使用极不均衡,翻转次数最多的总线与翻转次数最少的总线之间最大有近 20 万倍的差值。

整个总线系统是一个整体,单根总线的失效往往会影响整个系统的性能,因此绿色编译优化需要综合单根总线的峰值翻转次数和总的翻转次数,通过最小化式(1.3)获得更好的优化效果。本书后续部分将介绍在减少总线总翻转频度的同时,如何使各总线翻转频度尽可能均衡。

$$\frac{\sum_{t=0}^T \sum_{i=0}^{n-1} |b_{i,t} - b_{i,t+1}|}{T \times n} \quad (1.2)$$

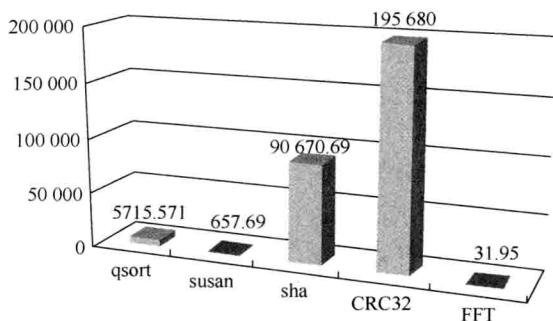


图 1.1 单根总线动态负载对比图

$$\alpha \times \frac{\sum_{t=0}^T \sum_{i=0}^{n-1} |b_{i,t} - b_{i,t+1}|}{T \times n} + \beta \times \frac{\max_{i=0}^{n-1} (\sum_{t=0}^T |b_{i,t} - b_{i,t+1}|)}{T} \quad (1.3)$$

其中, T 表示总的执行时钟周期; n 表示总线位宽; $b_{i,t}$ 表示第 i 根总线在时刻 t 的状态(其值为 0 或者 1); α 和 β 是两个权重因子, 用于权衡总的翻转次数与单根总线的峰值翻转次数对系统的影响。

2. 针对总线串扰的优化

总线串扰是指相邻总线之间在并行传输数据时出现的电容性耦合以及电感性耦合现象。这些耦合将产生干扰电流和干扰电压, 不但消耗额外的能耗, 而且影响传输数据的可靠性和稳定性, 增加额外的传输延迟。总线串扰的大小同总线长度、总线之间的距离以及总线翻转的模式有很大关系, 可分别用式(1.4)与式(1.5)表示其能量消耗以及传输延迟^[24~26], 即

$$E_{\text{total}} = \sum_{j=1}^n (1 + C_{\text{eff},j}) C_L \times V_{\text{dd}}^2 \quad (1.4)$$

$$\tau_j = k \times V_{\text{dd}} \times C_L \times C_{\text{eff},j} \quad (1.5)$$

其中

$$\Delta_{i,j}^{t,t+1} = \begin{cases} 0, & \Delta_i^{t,t+1} = 0 \\ \text{abs}(\Delta_i^{t,t+1} + \Delta_j^{t,t+1}), & \text{其他} \end{cases} \quad (1.6)$$

$$\text{abs}(a) = \begin{cases} a, & a \geq 0 \\ -a, & a < 0 \end{cases} \quad (1.7)$$

$$T_{\text{condition}}\{\text{exp}\} = \begin{cases} \text{exp}, & \text{condition} = \text{true} \\ 0, & \text{condition} = \text{false} \end{cases} \quad (1.8)$$

其中, n 表示并行传输数据的总线根数; C_L 表示总线与地线之间的电容; $\Delta_{i,j}^{t,t+1}$ 表示相邻总线 i 与 j 翻转值; $C_{\text{eff},j} = \text{abs}(\Delta_j^{t,t+1} (1 + T_{j>1} \{\lambda \times \Delta_{j,j-1}^{t,t+1}\} + T_{j<n} \{\lambda \times \Delta_{j,j+1}^{t,t+1}\}))$

表示第 j 条总线有效电容; $T_{\text{condition}\{\text{exp}\}}$ 中 condition 为布尔条件表达式, exp 为任意算术表达式; $\lambda = C_I / C_L$ 表示总线之间电容相对于总线本身电容的比值, C_I 表示相邻总线之间的串扰电容; $\Delta_j^{t+1} = b_{t+1,j} - b_{t,j}$ 表示第 j 条总线在相邻两个时钟周期 t 和 $t+1$ 的状态变化值; V_{dd} 表示工作电压。

由式(1.4)和式(1.5)可以看出, λ 值越大, 总线串扰越大, 导致的无效能源消耗也越大, 如何降低总线串扰对减少系统能源消耗有着重要影响。

随着半导体工艺的快速发展, 总线布局越来越密集, 总线之间的串扰也越来越大。当半导体工艺达到 $0.18\mu\text{m}$ 时, λ 值即将达到 6, 当半导体工艺为 $0.13\mu\text{m}$ 时, λ 值将超过 8^[27]。因此, 总线串扰已经成为了 CMOS 电路中不容忽视的重要方面。

为改善总线的工作环境, 降低总线能耗, 电路设计者在体系结构的设计上提出了一些减少总线串扰的方法, 主要方法有编码^[28]、屏蔽线(shielding)^[29,30]、门电路缩放(gate sizing)^[31,32]。此外, 还有一些方法通过调整总线空间布局, 使容易产生串扰的总线分布在不相邻的位置, 或者采用不均匀布线, 增加这些总线之间的距离^[33]。同所有硬件解决方案类似, 这些方法或多或少都需要额外的开销, 如编码方法需要额外的编/解码电路, 屏蔽线需要为增加的屏蔽总线分配额外的空间。因此, 除从硬件体系结构消除总线串扰外, 结合编译优化技术调整软件的执行方式, 是进一步降低总线串扰的必然选择。一些研究者已经开始在这方面展开研究。

Weng 等根据文献[30]中使用的选择屏蔽线的特点, 利用编译器后端的寄存器重命名技术, 改变指令总线上传输的数据, 使其更有利于发挥选择屏蔽线的功效, 减少总线串扰, 以进一步提高总线性能, 降低总线的能耗。其实验结果表明该方法能够在文献[30]的基础上节约 13% 的能耗^[34]。Kuo 等综合使用了三种后端编译优化方法。

① 利用指令调度调整指令执行序列, 减少操作码之间的总线串扰。

② 利用寄存器重命名技术, 通过调整相邻指令间操作的寄存器, 减少寄存器类操作指令操作数之间的总线串扰。

③ 利用 Nop 填充技术, 在存在 4C 类串扰的指令间插入冗余的 Nop 指令, 以进一步减少 4C 类总线串扰。

通过模拟实验, 该方法能够将 4C 类串扰从原来的 11.50% 减小到 0.52%^[35]。

由此可见, 结合编译优化能够有效地减少总线之间的串扰。但目前基于编译优化减少总线串扰的研究主要集中在简单的寄存器重命名, 并没有充分发挥编译器中丰富的数据流信息及具体指令的特点。如果能够结合这些信息, 我们将获得更好的优化效果。图 1.2 显示了一个简单的改进总线之间串扰的方法。

如文献[35]中图 1.2(a)所示指令序列(加粗部分表示有 4C 类串扰), 采用寄