学校编码：10384　　分类号_____密级_____
学　　号：23120090153689　　UDC_____

厦门大学

博 士 学 位 论 文

嵌入式系统的编译器设计及其关键技术研究

Research on Compiler Design and its Key Technologies for Embedded Systems

陈 本 彬

指导教师姓名：郭 东 辉 教 授
专 业 名 称：电 路 与 系 统
论文提交日期：2015 年 3 月
论文答辩时间：2015 年 5 月
学位授予日期：

答辩委员会主席：_____
评　　阅　　人：_____

2015 年　月

厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下，独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规范（试行）》。

另外，该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年　　月　　日

厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（　　）1。经厦门大学保密委员会审查核定的保密学位论文，于　　年　　月　　日解密，解密后适用上述授权。

（　　）2。不保密，适用上述授权。

（请在以上相应括号内打“ √ ”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年　　月　　日

摘　要

嵌入式系统处理器通常采用特定的指令集，其体系架构不具有通用性，因此，嵌入式系统的编译器开发无法满足不同嵌入式系统的软件开发需求，即使是通过移植方法来开发相应的嵌入式系统的编译器，也面临着可重定向方案设计等难题。针对嵌入式编译器的重定向难题，嵌入式系统对开发效率及高质量目标代码等多方面的需求，本论文就专用指令集嵌入式处理器（ASIPs）的编译器重定向、基于 Cache 的编译器优化及嵌入式多核静态调度等方面的理论和实践进行研究。

在对编译器的原理及理论基础进行介绍后，本论文首先针对 ASIPs 特点分析研究编译器可重定向及移植问题；接着给出能够实现编译器静态预测的方法，提高 Cache 复用率，解决“存储墙”瓶颈问题；最后针对嵌入式多核编译静态调度的负载不均衡问题，提出有效的静态调度方案。本论文的创新点主要体现在：

（1）提出针对 ASIPs 的重定向编译器的扩展方法。针对一款具有不规则架构且硬件资源有限的嵌入式 8 位微处理器及其扩展的标准 ANSI C 编程语言，以 GCC 编译器为基础，进行编译器前端的标识符扩展与语法树属性合并，和编译器后端目标描述文件（MD）与处理器硬件资源的差异性定义，设计出一套可重定向的编译器。

（2）提出基于 Markov 模型的节点频率预测方法。通过程序控制流图（CFG）构建的 Markov 模型，分析程序的特性，结合传统编译器静态启发式分支预测和基于 Markov 程序参数可调的预测方法，通过编译器静态的混合预测，更加精确的预测分支概率，计算程序节点频率（NF），以提高嵌入式 Cache 的利用率。

（3）提出一种基于评估函数的 OpenMP 静态编译调度方案。通过对多核调度、任务划分和负载均衡等问题的描述，基于对并行循环的划分定义及负载函数的映射模型，提出一种基于静态评估函数的调度方法，以解决编译器静态调度的负载不均衡问题，提高嵌入式程序的并行效率。

最后，实践开发了一套针对专用指令集系列商用 SoCs 的 C 语言 HCC 编译器，并集成相应编译器到本地集成开发环境（IDE）及实验室的 WEB-EDA 平台上。HCC 编译器已经投入市场，通过覆盖率等测试反馈，验证其编译稳定高效。

关键字：嵌入式编译器；ASIPs；节点频率预测；任务调度

ABSTRACT

There are different system architectures, specific instruction sets and tailored peripheral controllers in embedded System on Chips (SoCs). Because of the architectures versatility in embedded SoCs, the compiler development based on embedded system can't satisfy all the requirements for all the embedded processors. These result in the lack of embedded compiler and the difficulty of compiler transplantation and compiler maintenance in embedded system. For improving the generated codes quality and performance, this thesis discusses the retargetable compiler based on embedded Application-Specific Instruction-set Processors (ASIPs), the compiler optimization based on embedded Cache and the static scheduling of embedded multicore compiler, etc. These studies and practices could guide the quick porting of retargeting compiler especially for ASIPs and meet the performance requirements of embedded system.

After introducing the basic knowledge of the compiler and the latest research theories, this paper discusses special characteristics of ASIPs and the transplant and retargetting methods for ASIP with compiler extension firstly. And then, the "storage wall" bottleneck of Cache is described and an effective method to improve the accuracy of the compiler static prediction is proposed. Finally, based on the introduction of multi-core embedded processors, the compiler static scheduling scheme with load balance is put forward to adapt the parallel application of OpenMP. The innovation points of this paper are mainly embodied in follow:

（1）The High-performance C Compiler (HCC) and its specific extension and implementation for ASIPs is proposed. HCC compiler is a language C compiler based on the retargetable GCC compiler. Because of the specialized architectures, the compiler extension methods are proposed in HCC compiler to quickly implement a compiler for ASIPs. We extend the identifier and attribute with Abstract Syntax Tree (AST) for language-specific programming syntax of the compiler front-end, which is

the syntax of the extension standard ANSI C. And then, the machine-dependent classification of assembly generation for the specific embedded SoCs is designed and implemented. After finishing MD (Machine Description) of the compiler back-end, the HCC compiler is completed by retargeting GCC compiler for ASIPs.

（2）For improving the average accessing time of memory subsystem by raising the cache hit rate, the Nodes Frequencies (NF) prediction techniques of compiler-assisted Markov Parameters Tuning (MPT) are proposed for codes relocation. Different with the traditional scheme that provide the Fixed Heuristics Branch Probability (FHBP) to calculate NF for various kinds of programs, the Markov-based heuristics algorithm combining FHBP and MPT is adopted for improving program NF prediction to take advantage of the Markov probability matrix which is modeled from the Control Flow Graph (CFG) of function. It can get the prediction of branch probabilities more precision for compiler to impove codes relocation that is very useful to raise the cache hit rate.

（3）This paper proposes the a new static scheduling scheme based on evaluation function. The scheduling, load balancing, and scheduling overhead are described with formal methods, and based on definition of loops and the mapping of the corresponding load functions, a new static scheduling scheme of multi-core compiler based on evaluation function is put forward in order to solve the load imbalance problem. This new static scheduling improves the parallel efficiency of the embedded application, especially for multimedia application with a lot of loops.

Finally, we have developed a retargetable HCC C language compiler and the Integrated Development Environment (IDE) based on the hardware of a series for commercial specific instruction set SoCs and the open sources GCC. The compiler is also integrated into the WEB-EDA platform of our laboratory. According to the crossing contrasts and tests, conclusion can be drawn that the proposed compiler has excellent improvement of the generated assembly codes. The compiler has already put on the market.

Key words: Embedded Compiler; ASIPs; Node Frequencies Prediction; Scheduling

目　　录

TABLE of CONTENTS