

Research on
Reconfigurable Computing for
Multi-core System

面向多核系统的 可重构计算研究

胡 威 严力科 著



科学出版社

面向多核系统的可重构计算研究

胡 威 严力科 著

科 学 出 版 社

北 京

版权所有,侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书以集成可重构器件的多核系统为基础,提出面向多核系统的可重构计算框架,并以此为支撑开展相关的研究,给出对应的设计与验证分析。面向多核系统的可重构计算,从可重构器件片上面积管理与配置优化、可重构器件支持的软硬件线程、可重构多核处理器体系结构、多核架构上可重构任务的系统支持、约束驱动的可重构加速核性能资源优化、面向片上网络的扩展等方面进行了研究与探索。多核系统中所集成的可重构计算研究,通过可重构配置来实现计算任务的灵活处理,并通过配置出的硬件来完成特定的计算任务,提高片上可重构资源的利用效率,改善软硬件资源的利用方式,扩展了可重构多核处理器结构设计方法,改进了可重构任务的动态管理效率,实现了性能和可重构资源的平衡,探索了片上网络多核系统中的资源利用,充分发挥了可重构器件的灵活性和计算能力,从而提高多核系统的整体性能。

本书可供从事计算机系统结构、可重构计算和多核计算研究的科研人员、教师、工程师参考。

图书在版编目(CIP)数据

面向多核系统的可重构计算研究/胡威,严力科著. —北京:科学出版社, 2016.12

ISBN 978-7-03-050950-5

I. ①面… II. ①胡… ②严… III. ①微处理器-系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2016)第 285128 号

责任编辑:闫 陶 杜 权/责任校对:董艳辉

责任印制:彭 超/封面设计:苏 波

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>

武汉中科兴业印务有限公司印刷

科学出版社发行 各地新华书店经销

*

开本:B5(720×1000)

2016年11月第一版 印张:14 1/4

2016年11月第一次印刷 字数:280 000

定价:70.00 元

(如有印装质量问题,我社负责调换)

前 言

半导体技术的进步和应用对计算能力的需求,持续推动着计算机体系结构的发展。当前主流的多核处理器仍然是为通用计算任务设计,尽管单芯片上所集成的通用处理器核数量不断增加,仍然难以满足计算密集型的高性能计算需求。将定制的 ASIC 加速核集成到多核处理器上是解决某些特定应用需求的方法,能够满足部分应用所需的计算能力。但是由于 ASIC 加速核只能对特定应用进行加速,缺少足够的灵活性。而可重构器件具有定制化和可重配置的双重优势,既可以获得高于通用处理器核的计算性能,又可以通过重配置来获得灵活性,同时可重构器件在动态局部重构、多配置上下文结构等方面的发展也为其带来更高的扩展性和实用性。

具有可重构器件的处理器由通用处理器核与可重构器件共同构成。由于可重构器件所带来的灵活性和面向应用的高性能,目前对可重构计算,已经以“单核+可重构器件”的方式开展研究。当片上集成了多个处理器核时,可重构处理器的体系结构远较单核复杂,因此,以传统单核可重构处理器的可重构计算研究已经不能满足多核系统的要求。多核可重构系统中,多个通用处理器核与可重构器件之间的通信和交互方式与单核结构具有较大的区别。要对请求进行高效的影响,能够高效地完成计算任务,提高可重构资源的管理效率,就需要对多核可重构系统进行探索。此外,在多核以片上网络形式存在,还需要进一步探索以片上网络支持的体系结构中计算效率与可重构的设计。因此,本书以多核可重构体系结构的发展趋势为基础,研究如何在现有多核处理器体系中扩展集成可重构处理单元、在该扩展结构上可重构处理单元的运行支持管理和可重构任务管理调度,以及多可重构功能核实现面积和性能平衡的优化选取。本书主要从以下 6 个方面进行了研究。

(1) 提出了可重构器件片上面积管理与配置优化方法,建立了器件模型和任务模型,通过顶点的插入与删除来进行片上面积的管理,以较低的代价来实现硬件任务的快速面积分配与回收,提高可重构器件的利用效率。通过将配置过程流水化,来提高配置效率,从而提高系统性能。

(2) 提出了面向可重构器件的软硬件混合线程模型,将可分配到可重构器件执行的计算任务构建为硬件线程,通过库来进行硬件线程的创建与管理,通过硬件调度器来实现硬件线程的调度,从而能够更为高效地利用可重构器件,并实现软件

系统与可重构器件的协同。

(3) 建立了可重构多核处理器结构模型,通过分析可重构处理单元执行效率的影响因素,得到可重构多核处理器结构的设计原则,并在此指导下设计了一种可重构多核处理器结构,包括通用处理核的扩展、互连结构设计、可重构处理单元设计以及可重构处理单元管理器设计。

(4) 提出了执行与配置分离的处理模式,在提高可重构功能核的执行效率的同时避免过度频繁的管理调度,降低系统开销;并且在考虑可重构资源的单位面积利用效率的情况下,提出了3种可重构任务动态调度算法。

(5) 针对相同功能的可重构功能核存在多种实现的情况,提出了系统性能约束驱动的功能核实现优化选取方法,目的是达到性能和可重构资源的平衡,充分发挥功能核性能并避免资源浪费。

(6) 针对片上网络多核系统开展了扩展研究。提出了局部总线混合片上网络结构、片上网络中可重配置的连线结构以及面向低功耗的片上网络多线程映射算法,目的是通过对片上网络体系结构和线程映射的研究,探索在众核环境下计算性能和网络性能的改善和提升。

本书所开展的面向多核的可重构计算研究,以“多个通用核+可重构器件”为基本的体系结构,通过可重构器件的片上面积管理和配置优化、基于可重构的硬件线程、可重构多核处理器体系结构、可重构多核系统中的可重构任务系统支持、约束驱动的可重构加速核性能资源优化、基于片上网络的扩展等方面,对多核架构上的可重构计算进行了研究,给出了相应的解决方案,提升了多核可重构计算系统的性能,有效推动了可重构器件在多核处理器上的进一步研究和应用。

本书综合了相关专家学者的最新成果和本人的研究工作,但由于本领域技术内容丰富、发展迅猛,书中难免存在不足或疏漏之处,希望广大读者批评指正。

编 者

2016年9月

目 录

第 1 章 绪论	1
1.1 处理器体系结构的发展	1
1.2 可重构计算发展	3
1.3 可重构计算的优势	4
1.4 面向多核系统的可重构计算的研究意义	5
第 2 章 多核体系结构与可重构计算进展	9
2.1 多核体系结构研究进展	9
2.1.1 片上多处理器体系结构	9
2.1.2 片上互连结构	11
2.2 可重构硬件研究进展	14
2.2.1 可重构器件	14
2.2.2 动态重配置模式	16
2.2.3 多上下文可重构硬件	17
2.3 可重构处理器体系研究进展	18
2.3.1 单通用核可重构处理器	19
2.3.2 可重构多核处理器结构	25
2.4 软硬件任务运行时环境研究进展	26
2.4.1 可重构器件空间管理	26
2.4.2 软硬件任务模型	28
2.4.3 硬件任务调度管理研究进展	29
2.5 重配置代价优化研究进展	32
2.5.1 配置缓存	33
2.5.2 配置预取	34
2.5.3 局部可重构系统中重定位和碎片整理	34
2.5.4 配置压缩	35
2.6 面向片上网络多核系统的研究进展	35
2.6.1 片上网络基本拓扑结构	35
2.6.2 片上网络的结构优化研究进展	36
2.6.3 片上网络的 I/O 研究进展	37

2.7 目前研究存在的问题	38
第3章 面向多核系统的可重构计算研究概述	40
3.1 整体研究框架	40
3.2 研究内容概述	43
3.2.1 可重构器件片上面积管理	43
3.2.2 面向可重构计算的硬件线程	44
3.2.3 可重构多核处理器体系结构	45
3.2.4 多核架构上可重构任务的系统支持	46
3.2.5 约束驱动的可重构加速核性能面积优化	47
3.2.6 面向片上网络的扩展研究	47
3.3 性能分析	49
3.3.1 性能分析内容	49
3.3.2 实验与验证方法	50
3.3.3 系统模拟平台扩展	51
3.3.4 硬件加速核的验证平台	52
3.3.5 操作系统和应用程序修改	53
第4章 可重构器件空间管理与配置优化	54
4.1 可重构器件模型	54
4.2 可重构器件上的任务模型	56
4.3 空闲空间管理方法	57
4.3.1 候选位置	57
4.3.2 可用候选位置	59
4.3.3 任务的插入	60
4.3.4 任务的清除	61
4.3.5 算法描述	62
4.4 流水化配置优化	64
4.4.1 重配置叠加与流水	64
4.4.2 中间数据缓冲	66
4.5 验证与分析	68
4.5.1 分配质量分析	68
4.5.2 计算复杂度	70
4.5.3 流水化配置优化验证与分析	71
第5章 基于可重构计算的硬件线程	76
5.1 混合架构模型	76
5.2 硬件线程模型	77

5.2.1 软硬件混合多线程	77
5.2.2 软硬件线程运行时映射	79
5.3 硬件线程执行机制	80
5.3.1 硬件线程控制块	80
5.3.2 硬件线程执行器	81
5.3.3 软硬件线程的协同执行	84
5.4 硬件调度器	86
5.4.1 硬件支持的调度	86
5.4.2 硬件调度器设计	90
5.5 验证与分析	94
5.5.1 硬件线程的验证与分析	94
5.5.2 硬件调度器的验证与分析	98
第6章 可重构多核处理器结构设计	100
6.1 可重构多核处理器模型	100
6.1.1 可重构多核处理器结构模型	100
6.1.2 可重构处理单元执行效率	102
6.1.3 可重构多核处理器结构设计优化目标	103
6.2 一种可重构多核处理器总体结构设计	104
6.3 通用处理核结构扩展研究	105
6.3.1 通用处理核接口和指令扩展	105
6.3.2 互连结构	107
6.4 RPU 的结构设计	108
6.4.1 RPU 的 CLA 和 CCM	109
6.4.2 输入缓存的设计和组织	111
6.5 RPU 管理器设计	111
6.5.1 配置上下文查找单元	112
6.5.2 执行控制单元	115
6.5.3 选通控制单元	116
6.6 结构模型参数	117
6.7 可重构多核处理器结构的验证与分析	117
6.7.1 单任务实验验证与分析	117
6.7.2 多任务实验验证与分析	124
第7章 多核架构上可重构任务的系统支持	130
7.1 可重构计算资源抽象与处理模式	130
7.1.1 可重构资源抽象模型	130

7.1.2	RPU 和 RFC 的状态抽象	131
7.1.3	执行与配置分离处理模式	133
7.2	可重构任务的程序组织与执行	134
7.2.1	可重构任务的程序组织	134
7.2.2	硬件任务的执行机制	135
7.2.3	RFC 的配置构造	137
7.3	可重构任务的调度管理	139
7.3.1	可重构任务抽象模型	139
7.3.2	问题描述	141
7.3.3	多配置上下文结构上的多任务调度策略	142
7.3.4	调度策略评价指标	148
7.4	可重构任务管理调度的验证与分析	149
7.4.1	实验设置	149
7.4.2	运行性能结果与分析	149
7.4.3	请求命中率结果与分析	151
7.4.4	资源利用率结果与分析	152
7.4.5	重配置开销结果与分析	152
7.4.6	调度管理代价结果与分析	153
第 8 章	约束驱动的可重构加速核性能资源优化	155
8.1	约束驱动的 RFC 实现优化选取模型	155
8.1.1	系统性能约束因素分析	155
8.1.2	约束驱动的选取模型	157
8.2	约束驱动的 RFC 实现优化选取机制	158
8.3	带宽约束驱动的 RFC 选取优化	160
8.3.1	带宽约束驱动的优化选取	160
8.3.2	带宽约束驱动优化的案例研究	161
8.4	输入数据约束驱动的 RFC 选取优化	163
8.4.1	输入数据约束驱动的优化选取	163
8.4.2	输入数据驱动优化的案例研究	165
8.5	约束驱动的功能核选取验证与分析	168
8.5.1	带宽驱动的功能核优化选取	168
8.5.2	输入数据驱动的功能核优化选取	171
第 9 章	面向片上互连的扩展研究	175
9.1	局部总线与片上网络混合结构	175
9.1.1	混合结构设计	175

9.1.2 线程调度支持	176
9.2 基于可配置连线的片上网络结构	178
9.2.1 基于可配置连线的片上网络结构设计	178
9.2.2 支持 I/O 的结构设计	181
9.2.3 重配置与映射	182
9.3 低功耗片上网络多线程映射算法	183
9.3.1 片上网络功耗模型	183
9.3.2 低功耗映射算法	185
9.3.3 在线低功耗映射	187
9.4 实验与验证	188
9.4.1 混合片上网络结构的验证与分析	188
9.4.2 可重配置连线片上网络的验证与分析	190
9.4.3 低功耗映射算法的验证与分析	194
参考文献	199

第 1 章 绪 论

由于半导体工艺的进步及其应用的推动,处理器体系结构已经取得很大成功。尤其是多核技术的出现,使得处理器的性能和效率得到了提升。在此基础上,以多核技术为基础的计算系统提供了更多的计算形式,具有发展潜力。可重构计算兼有近似 ASIC 的高性能和通用处理器的灵活性的特点,符合未来处理器结构设计的要求,因此,集成可重构处理器(RPU)的处理器体系成为未来的重要发展方向。本章对处理器体系结构的发展进行了介绍,并对可重构计算的特点和优势进行了分析,提出了面向多核系统的可重构计算研究。

1.1 处理器体系结构的发展

进入 21 世纪以来,半导体工艺的进步和高性能计算的需求推动了计算机系统的迅猛发展,处理器的体系结构发生了巨大变化。作为计算机核心部件的处理器,其性能的提升在很大程度上归功于半导体工艺的进步和体系结构的发展。半导体工艺的进步和体系结构的发展一直是相互促进的关系,前者是后者的基础,后者又推动着前者。体系结构发展上的每一次变化都伴随着芯片上集成晶体管数目的突破,可以说是晶体管数目的“量变”导致的“质变”。

以市场主流的 Intel 处理器为例,纵观过去 40 年间半导体工艺和处理器体系结构的历史,基本上沿着如图 1.1 所示的路线发展。戈登·摩尔(Gordon Moore)在 1965 年提出并于 1975 年修正的摩尔定律指出了半导体工艺发展的规律和趋势,即芯片上集成的晶体管数量将每两年翻一番^[1]。半导体工艺的发展一直遵循着摩尔定律,芯片线宽从 1971 年 4004 微处理器的 10 μm 缩小到 2009 年的 32 nm, 2011 年缩小到 22 nm,未来很有可能进一步缩小到 14 nm 以内。随着工艺制程的提高,处理器芯片上所集成的晶体管数目不断增加,从 4004 的 2000 多增加到了 Itanium Tukwila 核心的 20 亿。在半导体工艺进步的同时,体系结构也在持续深入发展,CPU 的设计越来越复杂而精密,在图 1.1 中,把未采用超标量架构的处理

器称为简单单核处理器,随着晶体管数目的增加,简单单核处理器从 4 位发展到 32 位,然后采用流水线技术,并且在芯片内集成 L1-Cache;复杂单核处理器是指采用了超标量架构的单核处理器,在逐渐增加的晶体管支持下,超标量技术、片内 L2-Cache、多媒体指令集扩展及同时多线程等体系结构设计逐步被采用,提高了处理器的性能,增强了浮点、矢量运算和多媒体等的处理能力^[2]。

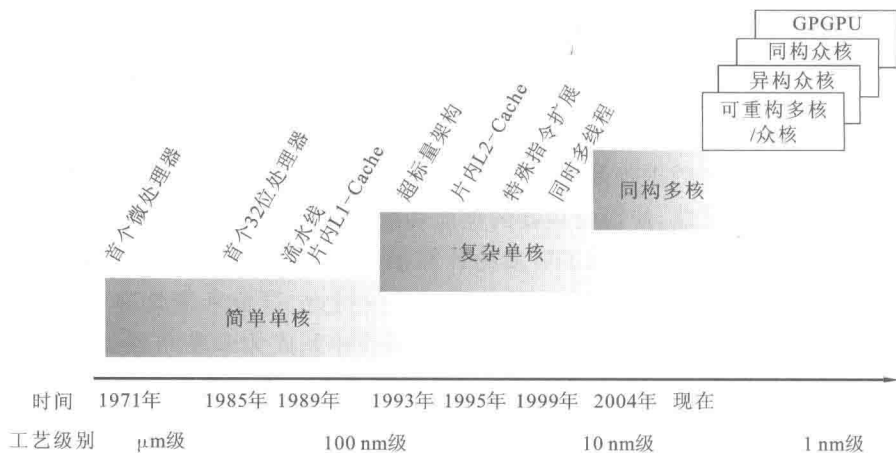


图 1.1 处理器体系结构大致发展路线

单核处理器体系结构的发展主要集中于指令级并行性 (instruction-level parallelism, ILP) 的充分开发与利用^[3],但是 ILP 技术的深度采用已经使得单核处理器设计复杂度越来越高,片上晶体管数目和功耗的增长比性能的增长要快得多,而指令间的可并行性是很有限制的,其结果是采用 ILP 技术来提高处理器性能变得越来越困难^[4]。因此,整体的发展趋势正逐步从单核体系结构的研究转向多核 (chip multiprocessor, CMP 或者 multi-core),利用线程级并行 (thread-level parallelism, TLP) 来继续提高性能^[4-6]。早在 1996 年,斯坦福大学的 Olukotun 等就提出并实现了单芯片多核的 Hydra 微架构^[7],但直到 2005 年初,主流通用处理器厂商 Intel 和 AMD 才相继推出双核处理器^[8],此后主流处理器进入同构多核时代,并迅速发展,片上处理核数逐渐增加,如 Intel 的 8 核 Xeon 处理器^[9]、IBM 的 8 核 Power7 处理器^[10]、Sun 的 8 核 UltraSPARC T2 处理器^[11]以及中国科学院计算技术研究所的 4 核龙芯 3A 处理器^[12]等。

多核体系结构延续了处理器的性能提升,获得了很大的成功,但是人们对性能的追求是没有极限的,应用的需求促使处理器性能的继续提升。未来处理器体系结构的发展方向是现在体系结构研究领域的热门话题,目前看来,未来可能的几种发展趋势包括如下。

(1) 同构多核^①,在单一芯片中集成超过 32 个相同的处理核,如 Intel 的 48 核处理器^[13]和 80 核处理器^[14]、Tilera 的 64 核处理器^[15]等。

(2) 异构多核,在单一芯片中集成多个不同结构或不同计算能力的处理核,如 IBM 的 Cell^[16]集成 1 个 PowerPC+8 个辅助处理核,AMD Fusion^[17]采用 CPU+GPU 结构。

(3) GPGPU,即通用图形处理器,利用 GPU 进行通用计算和加速计算,如 NVIDIA 的 Tesla^[18]。

(4) 可重构处理核,在单一芯片中集成通用处理核(general purpose core, GPC)与可重构逻辑,可重构逻辑可用于连接数百个简单处理单元,如 MIT 的 RAW^[19],或可重构加速核,如加州大学的 Warp^[20]等。

1.2 可重构计算发展

可重构计算(reconfigurable computing)的概念最早由 Estrin 等在 1963 年提出^[21],但是现在的概念与当时已有较大不同,目前学术界所常用的可重构计算是指:系统具有某种形式可编程性的硬件,可通过一系列物理控制点定期定制硬件的功能,从而可以使用相同的硬件执行不同的应用^[22,23]。可重构计算有望填补硬件计算(基于 ASIC 的计算)和软件计算(基于通用处理器的计算)之间在性能与灵活性上的鸿沟,从而在获得高于软件计算的性能的同时保持高于硬件计算的灵活性^[22]。

可重构计算的器件通常由计算单元(computational elements)阵列和连线资源(routing resources)构成,两者都是可编程的。计算单元又称为逻辑块(logic block),它的功能由一定数量的配置位所决定,逻辑块之间的互连由连线资源所决定^[23]。逻辑块实现简单的逻辑功能,经过可配置连线的连接从而实现复杂的定制功能。根据逻辑块粒度的大小,可重构器件可以分为细粒度结构和粗粒度结构,一些更细化的分类也将粒度大小分为细粒度、中等粒度、粗粒度和超粗粒度^[24]。最常见的可重构硬件当属 FPGA(field programmable gate arrays),有些文献直接将基于 FPGA 的计算等同于可重构计算^[25],FPGA 就是一种细粒度可重构器件。

一般可重构计算系统都采用可重构硬件和通用微处理器结合的形式,通用处理器执行可重构硬件无法高效完成的操作,如数据依赖的控制、存储访问等,而计算密集的程序热点则被映射到可重构硬件上完成。根据通用微处理器和可重构硬件的耦合方式,可重构硬件在计算系统中大致可以分为 4 类:可重构功能单元、可

^① 本书不细分多核与众核,统一称为多核。

重构协处理器、可重构附属处理单元和可重构的独立处理单元。

在商用可重构硬件中, FPGA 最为成熟普遍, 而它通常作为独立的芯片甚至板卡的形式存在, 所以在现有商用可重构计算系统中, FPGA 的使用最多, 且多数是以上述后两种耦合结构存在的, 如作为附属处理单元的有 Intel QuickAssist 的 FSB-FPGA 加速系统^[26] 以及 XtremeData 的 FPGA 加速平台^[27] 等, 作为独立处理单元的有 SRC-7 可重构超级计算机^[28] 等。

对于上述前两种耦合结构, 基于单核处理器的结构的研究很多^[20,29,31], 但是由于过去半导体工艺的限制, 通用处理器性能有持续提升的空间以及可重构计算应用开发的不便等原因导致可重构处理器一直未能广泛使用。

1.3 可重构计算的优势

一种计算形式的特性可以从两个方面来进行描述: 灵活性和性能^[24]。从这两个方面进行考虑, 以性能作为横坐标、灵活性作为纵坐标, 常见的计算形式的特性可以大致表示为图 1.2, 其中: 基于通用处理器的通用计算的灵活性最高, 但其能开发利用的并发性有限, 性能相对最低; 而应用定制计算是为特定应用定制并优化的, 性能相对最高, 但是, 也正是由于其只为特定应用定制, 所以灵活性最差。这两个极端在性能和能耗方面的效率都相差百倍^[32]。在这两个极端之间, 有多种计算形式, 如图 1.2 中所示的领域定制计算和可重构计算, 在这些计算形式中, 可重构计算在灵活性和性能方面有望弥合应用定制计算和通用计算之间的鸿沟^[24]。

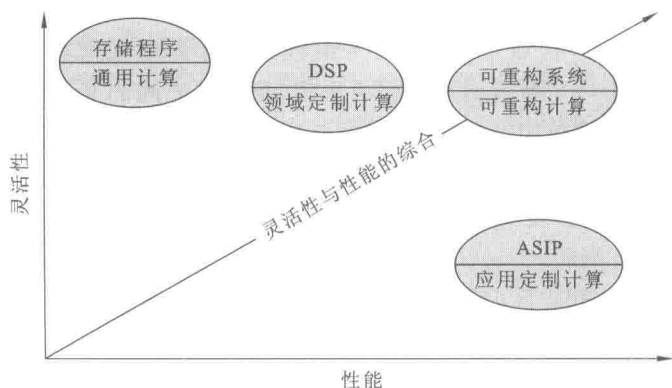


图 1.2 不同计算形式的灵活性与性能

可重构计算使用可重构硬件针对不同的应用进行定制, 可以充分利用从指令级到任务级间多个级别的并行性^[25], 从而达到接近 ASIC 的性能; 并通过运行时

重构(run-time reconfiguration, RTR)对可重构硬件进行电路功能的重配置,从而保持着接近软件的灵活性。因此,可重构计算相对于两个极端都具有更高的性价比,而与通用计算相比,在不同应用中取得了显著的性能提升且具有更低功耗^[33],如入侵检测^[34]、模式匹配^[35]、数值分析^[36]、生物信息^[37]等。

处理器体系结构设计的一个主要目标是在追求面向应用领域的高性能和高效率设计的同时,保持处理器的可编程性和灵活性,可重构计算在这方面有着其他计算形式无可比拟的天然优势。多核处理器早已成为当前的主流,成功延续了摩尔定律,而集成多通用核与可重构逻辑的可重构处理器将成为未来处理器结构的重要发展方向^[38]。

1.4 面向多核系统的可重构计算的研究意义

本书研究面向对性能和灵活性都有较高要求的计算系统,这些计算系统的用户需求可能随时发生变化,因此系统需要经常地针对用户的请求重新配置高性能的处理能力,如服务器、云计算终端等。对于这种应用需求,其他计算形式虽各有优势,但也存在着不足之处。同构多核由于 GPC 的通用性设计,性能不高,而多核处理器采用稍为简单的处理核以达到更高的能效比^[39],更加难以满足一些应用的性能需求,尤其是线程级并发度不高的应用。此外,由于存储墙等因素的存在,即使将来处理器片上集成更多的同构核,也很难继续提高性能,如美国桑迪亚国家实验室的一份报告指出,处理核个数超过 8 个后,处理器性能几乎没有提升,而到 16 个之后,性能不升反降^[40]。

异构多核及一些多核处理器以协处理器或处理单元的形式集成了定制的加速处理核来为特定的应用提供更高的性能,如 UltraSparc T2 集成了加密处理单元^[41],IBM Wire-Speed Power 集成 XML 加速核、正则表达式处理核和加密核^[41]等。这些 ASIC 的加速处理核虽能达到很高的性能,却缺乏灵活性,无法加速定制以外的应用。GPGPU 也可以达到较高的性能,但计算功耗较大,能效比不高,此外,GPGPU 是面向图形图像应用设计的,在很多应用上其性能不如可重构硬件的实现^[42,43]。

可重构硬件可以达到接近 ASIC 的性能,又保持了足够的灵活性,因此,在现有处理器中集成可重构处理核可能是更为合适的方法^[32]。可重构多核处理器可以更充分地利用从 ILP 到 TLP 的并行性^[38,44],并且,由于消除了频繁的取值和连续指令之间数据的存储带宽需求^[45],基于可重构硬件的加速核有利于缓解“存储墙”效应。

此外,随着多年的发展,目前可重构硬件的密度已经足够实现较复杂的功能并

且达到较高的工作频率,动态局部重构技术^[46,47]、多配置上下文结构^[48-51]等技术的研究与发展带来了可重构计算更高的灵活性和实用性。所以,本书认为多核处理器中集成可重构硬件是未来体系结构发展的一个重要方向。

可重构器件是重构的基础。在重构过程中,需要找到可以利用的区域来进行新的计算功能的实现。这一过程也需要花费较多的时间,不利于提高可重构计算的效率。因此,就需要首先开展可重构器件空间的管理研究,实现在线的空间管理与重构支持。这既包括了如何对空闲空间进行高效的管理,也包括了如何为在线的重配置的需要快速地寻找到合适的空闲空间并进行分配。这是提高可重构核处理效率的重要基础。

片上的可重构资源为程序的运行提供了新的方式。传统的多线程以软件的形式在处理单元上执行。在可重构资源的支持下,线程可以改变存在形式,以硬件线程的方式来执行。从而提高线程的执行效率。因此,需要研究以片上可重构资源为基础,硬件线程的设计方法及其执行机制;此外,由于受到资源数量的限制,软件线程和硬件线程共存,也需要对软硬件线程的数据共享方式、软硬件线程的调度等进行研究。

在现有的可重构处理器研究中,大多都是基于单核的通用处理器架构,对集成多 GPC 与多可重构处理单元(reconfigurable processing unit, RPU)的体系结构研究较少,由于设计目标或当时硬件技术的约束,之前的结构都存在着一些限制,例如,可重构资源不能动态合并、RPU 不能被各通用核共享等,因此,目前亟待在通用多核架构下进行可重构多核处理器结构研究,包括 GPC 与 RPU 的互连、可重构器件的空间管理、动态局部重构和多配置上下文技术的高效应用、多核互连结构的扩展等。

可重构任务的系统支持和管理调度一直是研究的重点,从这个角度看,多核架构与单核架构的最大不同在于:多个 GPC 可能在同一时刻发起多个 RPU 的调用请求,请求的次数和频率将比单核架构有较大提高。因此,还需基于多配置上下文的可重构多核处理器结构开展可重构任务支持和管理调度研究,包括如何高效地响应 RPU 的调用请求并执行,如何高效地管理可重构资源,如何调度可重构任务,以达到提高性能的同时,控制甚至降低系统的调度和重配置等开销。

为了适应不同数量的可重构资源,一些可重构功能核(reconfigurable functional core, RFC)会有多个具有不同的可重构资源需求和处理能力的实现。在运行时,系统性能的上限将会由一个或多个因素约束,如果约束性能的因素不是 RFC 处理能力,那么在多个 RFC 实现中选取更高性能的实现不仅不会提升性能,在多核多任务情况下还会导致资源浪费,影响系统整体性能的提高。在此情况下,如果适应性地选择与约束的最高性能相匹配的 RFC 实现,在不损害整体性能的同时却能节约可重构资源。因此,还有必要在性能约束驱动下,面向面积资源与性能

的平衡进行 RFC 选取的研究。

随着多核技术的不断发展,片上互连的结构也在不断发展。开展面向多核系统的可重构计算研究,需要对片上互连结构进行有效的扩展。尤其是随着片上处理器核数量的不断增加,传统的总线方式已经逐渐难以支持大量的通信需求,片上网络的核间连接和通信方式正在快速发展。对片上网络进行有效的探索,是进一步开展面向多核系统的可重构计算的重要支撑。

本书关注未来处理器体系结构中可重构计算资源的有效使用,从可重构资源管理的角度,开展可重构器件的空间管理和分配方法研究;从计算和控制等角度出发,开展软硬件线程与硬件调度器的研究;从体系结构角度出发,研究在通用多核处理器结构上多 RPU 的扩展、动态局部重构和多配置上下文结构的利用;在此扩展的可重构多核处理器结构基础上,研究多核多任务场景下可重构任务的配置和处理模式、可重构任务的管理调度;在同一功能的 RFC 存在多个不同实现的基础上,面向性能和可重构资源的平衡,研究性能约束驱动的 RFC 实现优化选取;对片上网络的扩展进行研究,为可重构资源的有效利用提供支持。

面向多核处理器体系结构,研究可重构器件的高效利用方法,在通用多核处理器结构上扩展 RPU,围绕可重构逻辑资源的高效管理,对扩展结构上的 RPU 执行机制和管理调度进行优化,对片上网络的扩展结构进行研究,进而提高系统的整体性能。本书主要在以下六方面进行了研究。

(1) 首先为可重构器件建立了器件模型和任务模型,提出以可重构器件上的 CLB 矩阵为基础,建立通过矩阵中的顶点来进行空闲空间管理的方法,实现高效率的面积分配和回收。通过将可重构器件上的计算任务划分为配置和执行两个阶段,并进行流水化配置优化,实现配置与执行的叠加,从而提高系统的性能。

(2) 首先分析了软硬件混合模式下的系统架构,建立了硬件线程模型,提出了硬件线程的设计方法与软硬件线程运行时的映射机制,通过软硬件线程的协同来提高软硬件线程的并行效率,提高硬件资源的利用率。提出了硬件调度器的设计机制,通过硬件调度器来实现硬件线程的高效调度,进一步改善软硬件线程的调度效率。

(3) 首先为可重构多核处理器建立结构模型,在此模型基础上以提高可重构处理单元的执行效率为目的,提出了体系结构的优化设计原则。然后根据此原则提出了一种可扩展的可重构多核处理器体系结构设计,利用了动态局部重构技术和多配置上下文结构,通过灵活的 GPC 与 RPU 互连结构,允许 GPC 间共享 RPU,以及合并多 RPU 以实现更复杂的功能。还设计了 RPU 的硬件管理器,在体系结构上支持 RFC 的高效调用执行和管理。

(4) 在上述可重构多核处理器体系结构上,进行了 RPU 的系统支持研究,主要内容包括 RPU 的执行机制研究和多上下文结构下的可重构资源管理和调度研