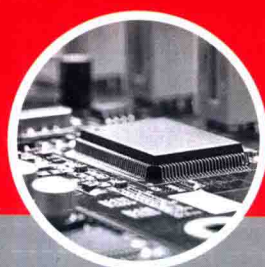


大学计算机学科学术研究进展系列丛书

片上软件系统

SOFTWARE SYSTEM ON CHIP

胡 威 严力科 著



科学出版社

大学计算机学科学术研究进展系列丛书

片上软件系统

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

本书以片上集成的可编程存储器为基础,提出片上软件系统的理论,并给出相应的设计与性能分析。片上软件系统的核心思想是,将以内存为基础来支撑软件系统的运行,转移到以片上集成的可编程存储器为基础来支撑软件系统的运行。片上可编程存储器具有访问速度快、功耗低的特点,能够有效地避免由于“内存墙”的存在所造成的高访问延迟、高功耗的问题。片上软件系统能够以片上可编程存储器的特点为基础,提升嵌入式操作系统的效率,加快程序的执行速度,减少消耗在存储器访问上的时间,降低因存储访问带来的能耗,改进嵌入式软件系统的运行效率,从而提高嵌入式系统的整体性能。

本书可供从事嵌入式存储系统、嵌入式系软件、嵌入式系统节能计算等方面研究的科研人员、教师、学生、工程师参考。

图书在版编目(CIP)数据

片上软件系统/胡威,严力科著. —北京:科学出版社, 2015. 11

(大学计算机学科学术研究进展系列丛书)

ISBN 978-7-03-046214-5

I. ①片… II. ①胡… ②严… III. ①程序设计 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2015)第 262648 号

责任编辑:张颖兵 杜 权/责任校对:董 丽

责任印制:高 嵘/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>

武汉市新华印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

开本:B5(720×1000)

2015 年 10 月第 一 版 印张:8 1/4

2015 年 10 月第一次印刷 字数:175 000

定价:45.00 元

(如有印装质量问题,我社负责调换)

前 言

半导体技术与计算机技术的发展推动着嵌入式系统的快速发展。随着嵌入式系统应用领域的不断扩大和应用深度的扩展,在性能、实时性与能耗等方面提出了更高的要求。嵌入式系统的软、硬件需要协同进行高效率的设计,来满足应用的需求。随着以 SoC 和 MPSoC 为基础的嵌入式系统研究不断深入,通过提高片上面积的利用效率,来集成更为丰富的器件,从而获得更好的性能;利用片上高性能器件来提升嵌入式系统的整体性能是当前嵌入式系统研究的重点之一。

嵌入式系统由于其面向应用领域定制的特点,往往对性能和功耗具有较为严格的要求。因此,如何提高嵌入式系统的性能、降低其功耗就成为嵌入式系统研究的重要方向。其中如何降低由于“内存墙”的存在所造成的访问延迟和高功耗是关键,而片上可利用面积的增加使得在嵌入式处理器上集成存储器成为解决这一问题的方法。片上可编程存储器有利于缩小处理器核与内存之间的访问延迟,提高系统的整体性能。通过软件控制的方法,可以将代码和数据存放在片上可编程存储器中,提供给处理器使用,支撑系统的运行。由于片上可编程存储器具有软件可控制、响应速度快、占用片上面积小、能耗低的特点,在嵌入式系统中得到了越来越广泛的应用。

本书对片上软件系统进行系统性的论述。单核嵌入式处理器与多核嵌入式处理器具有明显不同的硬件特点,所继承的片上可编程存储器也具有不同的配置和访问特点。针对集成了片上可编程存储器的芯片,片上软件系统进行了两种不同硬件平台上的研究,分别是基于单核嵌入式处理器的片上软件系统和基于 MPSoC 嵌入式处理器的片上软件系统。在理论研究的基础上,也分别进行性能的分析。片上软件系统的主要特点如下:

第一,片上软件系统以片上可编程存储器为硬件支持,构建了片上的嵌入式操作系统及其软件优化。与传统的嵌入式软件相比,由于片上软件系统将软件核心部分从片外存储器以及片上硬件控制存储器转移到了片上可编程存储器,降低了访问片外存储器所带来的延迟与功耗,减少了访问片上硬件控制存储器所带来的功耗。

第二,以片上软件系统为核心的嵌入式操作系统优化。设计了片上软件系统的调度算法,将片上可编程存储器与其他资源的状态加入到调度算法当中,从而使得调度算法能够根据片上可编程存储器和其他资源的状态来进行分组的调度;将

进程调度模块分配到片上可编程存储器上运行,从而加快进程调度模块的执行速度;将嵌入式操作系统微内核化,对微内核实行构件化,从而可以将微内核或者微内核的部分构件运行在片上可编程存储器上,提高操作系统的运行效率。

第三,以片上软件系统为中心提出了片上可编程存储器的多道程序共享。通过对嵌入式程序进行分析,生成存储对象。这些存储对象在运行时,可以被调度到片上可编程存储器上,利用片上可编程存储器的特点加快程序的运行速度,减少程序运行时的能耗。

第四,面向 MPSoC 体系结构的片上软件系统。对于传统的嵌入式程序,通过编译分析析取存储对象。通过操作系统专门的片上可编程存储器管理器在多道程序之间进行协同,达到多道程序共享片上可编程存储器的目的。通过对程序的线程化,提高程序的并行性,同时由于线程流水,存储对象将会相对稳定的驻留在片上可编程存储器上,从而实现对程序运行加速的目标。此外,在片上软件系统中引入了片上事务存储,利用片上可编程存储器来实现片上软件系统中的事务存储,同样可以提高系统的性能。

对于片上软件系统的性能,本书分别在嵌入式硬件平台上和模拟平台上的实验得到了验证:

一是在嵌入式硬件平台下验证了面向单核处理器的片上软件系统。通过多个测试程序的运行,对片上软件系统及其主要技术点进行了性能验证和分析。实验结果表明,面向单核处理器的片上软件系统能够有效地提高嵌入式系统的性能,降低系统运行时的能耗。

二是在模拟平台下对面向 MPSoC 的片上软件系统进行了验证。通过多个测试程序并行来验证 MPSoC 下多道程序共享;通过线程化的测试程序来验证多线程优化;将多道程序共享与多线程优化结合进行了验证。实验结果表明,在 MP-SoC 环境下,片上软件系统能够提高程序的运行速度,提升系统的性能。

本书关注于嵌入式软件系统的性能提升和能耗优化,以 ScratchPad Memory 等片上可编程存储器为硬件基础,提出片上软件系统的思想,通过将嵌入式软件系统从内存迁移到片上可编程存储器,从访问内存转变为访问片上的可编程存储器来提高数据获取速度、降低软件系统的能耗,提升嵌入式系统的性能。本书以片上软件系统为中心,主要进行了以下的研究:

第一,片上软件系统的整体设计框架。片上软件系统的设计思想,是以高集成度的 SoC/MPSoC 和片上可编程存储器为硬件基础,进行嵌入式软件系统的设计。本书研究了片上软件系统的整体设计框架,指出在基本硬件基础上,片上软件系统的主要构成,并综合分析片上软件系统的性能。

第二,面向片上可编程存储器的嵌入式操作系统优化。通过对嵌入式操作系统中调度算法的改进,对嵌入式操作系统的进程调度模块进行优化,将进程调度模块分配到片上可编程存储器上运行;将嵌入式操作系统微内核化,对微内核实行构

件化,从而提高嵌入式操作系统的运行效率。

第三,片上软件系统中多道程序共享片上可编程存储器。通过对嵌入式程序进行分析,生成存储对象。这些存储对象在运行时,可以被调度到片上可编程存储器上,利用片上可编程存储器的特点加快程序的运行速度,减少程序运行时的能耗。

第四,面向 MPSoC 体系结构的片上软件系统。对于传统的嵌入式程序,通过编译分析析取存储对象。通过操作系统专门的片上可编程存储器管理器在多道程序之间进行协同,达到多道程序共享片上可编程存储器的目的。通过对程序的线程化,提高程序的并行性,同时由于线程流水,存储对象将会相对稳定地驻留在片上可编程存储器上,从而实现对程序运行加速的目标。

本书所提出的片上软件系统,充分利用片上可编程存储器的特点,设计片上软件系统的框架,通过嵌入式操作系统、多道程序共享、多核嵌入式系统的优化,来实现片上软件系统对嵌入式系统的整体性能优化。本书的研究内容能够有效地通过基于片上可编程存储器的优化,提升嵌入式系统的整体性能,降低功耗推动片上可编程存储器在嵌入式系统中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。

胡 威

2015 年 8 月 18 日

目 录

第 1 章 绪论	1
1.1 嵌入式系统的发展	1
1.2 SoC 的发展	2
1.3 嵌入式系统中的片上存储器	3
1.4 片上软件系统的研究意义	4
第 2 章 片上可编程存储器	6
2.1 片上可编程存储器概述	6
2.1.1 片上可编程存储器的特征	6
2.1.2 集成片上可编程存储器的单核处理器结构	8
2.1.3 集成片上可编程存储器的 MPSoC 结构	10
2.2 嵌入式处理器的片上可编程存储器优化	12
2.2.1 不可重叠的片上可编程存储器优化	13
2.2.2 可重叠的片上可编程存储器优化	16
2.2.3 数组划分和循环变换的优化	19
2.2.4 面向 MPSoC 的片上可编程存储器优化	20
2.3 片上软件系统解决的问题	22
第 3 章 片上软件系统概述	24
3.1 片上软件系统整体框架	24
3.2 主要技术要点概述	26
3.2.1 片上软件系统中的嵌入式操作系统优化	26
3.2.2 多道程序共享片上可编程存储器	27
3.2.3 面向 MPSoC 的片上软件系统	27
3.3 片上软件系统性能综合分析	28
3.3.1 单核处理器硬件平台	29
3.3.2 多核模拟平台	31
3.3.3 软件环境与测试程序	32
3.3.4 性能分析	33
第 4 章 片上软件系统中的嵌入式操作系统	38
4.1 片上软件系统调度算法	38
4.1.1 任务模型	38

4.1.2	资源模型	38
4.1.3	任务分组与组内调度	40
4.1.4	调度算法设计	41
4.2	片上软件系统中的进程调度模块	44
4.2.1	进程调度代码与数据的组织	44
4.2.2	片上可编程存储器的空间划分	46
4.2.3	进程调度模块的重分配	47
4.3	片上软件系统中的微内核	50
4.3.1	片上微内核	50
4.3.2	资源封装	52
4.3.3	微内核构件化	53
4.4	性能分析	55
第 5 章	片上软件系统中的多道程序共享	57
5.1	多道程序共享方式	57
5.2	程序的编译分析	59
5.2.1	分析流程	59
5.2.2	控制流图分析	60
5.2.3	存储对象生成	63
5.3	片上可编程存储器的管理	66
5.3.1	虚实地址映射	66
5.3.2	片上可编程存储器的存储组织	66
5.3.3	片上可编程存储器运行时管理	69
5.4	性能分析	71
第 6 章	面向 MPSoC 的优化方法	75
6.1	MPSoC 的片上可编程存储器组织	75
6.1.1	处理器核组的划分	75
6.1.2	片上可编程存储器的组织	76
6.2	多道程序共享 MPSoC 的片上可编程存储器	78
6.2.1	程序的编译分析	78
6.2.2	核组调度与分配	81
6.2.3	片上可编程存储器管理	82
6.3	共享片上可编程存储器的线程调度优化	84
6.3.1	程序的多线程改进	84
6.3.2	片上可编程存储器的多线程优化	86
6.4	性能分析	89
6.4.1	多道程序共享优化性能分析	89

6.4.2 多线程优化性能分析	91
第7章 片上软件系统中的事务存储	92
7.1 片上事务存储	92
7.1.1 片上事务存储的定义	92
7.1.2 基本事务模型	94
7.2 片上可编程存储器的管理	96
7.2.1 基本存储管理组织结构	96
7.2.2 片上事务存储管理器	97
7.3 嵌套的片上事务存储	98
7.3.1 冲突仲裁策略	98
7.3.2 片上事务的嵌套方式	100
7.3.3 片上事务存储嵌套模型	102
7.3.4 嵌套的片上事务接口定义	103
7.4 性能分析	104
后记	107
参考文献	108

第 1 章 绪 论

随着计算机技术的发展,嵌入式系统的使用越来越广泛,应用环境对嵌入式系统的性能、实时性和能耗等方面的要求也越来越高。而通过 SoC 和 MPSoC 技术,在片上集成的存储器对提高嵌入式系统的性能有重要的作用;尤其是片上可编程存储器的出现,有利于减少程序运行时间,提高性能,减少能耗。本书首先介绍嵌入式系统的发展与片上可编程存储器的基本特点,以对片上可编程存储器的分析为基础,提出片上软件系统的设计思想。

1.1 嵌入式系统的发展

嵌入式系统出现于 20 世纪 70 年代,是最为常见的计算系统。随着半导体技术和处理器技术的不断发展,嵌入式技术日新月异,各类嵌入式系统得到了广泛的应用,包括工业领域和生活领域。嵌入式系统在不同的工业领域中的设计与应用已经超过 40 年,包括航空航天、铁路、能源和工业控制等领域和方向^[1]。与此同时,由于嵌入式系统在性能不断提高的同时成本也大幅度地下降,它也作为日常使用的电子设备,在汽车、家电和移动通信等领域得到了广泛的应用^[2],尤其是随着移动互联网和物联网等网络的发展,嵌入式系统更是成为必不可少的基础设备。嵌入式系统的使用远远超过了各种通用的计算机系统,在微处理器产品中,99%使用的是嵌入式系统^[3-4]。

从系统角度来看,嵌入式系统是以应用的需要为基础,采用计算机技术,对软硬件进行裁剪,从而满足定制要求的专用计算机系统^[5],具有面向用户、面向产品和面向应用定制的特点。嵌入式系统由于受到应用与应用环境的限制,在功能、可靠性、成本、体积和功耗等方面具有严格的要求,是软件与硬件一体化的计算系统。

通常嵌入式系统的结构可以分成嵌入式硬件和嵌入式软件两大部分。其中,嵌入式硬件又包括嵌入式处理器和嵌入式外围设备;而嵌入式软件则包括了嵌入式操作系统、嵌入式应用软件和开发工具。各种类型的嵌入式处理器是嵌入式系统的核心部件^[6],嵌入式系统的主要数据处理工作由嵌入式处理器完成,它决定了嵌入式系统的性能。嵌入式外围设备包括了 DRAM、LCD、键盘、音视频和传感器

等,具有很强的定制性。面向不同应用领域的嵌入式系统往往具有不同的嵌入式外围设备。嵌入式系统通过嵌入式外围设备来获得数据,并做出反馈。嵌入式操作系统负责控制与管理嵌入式系统的硬件资源,并向嵌入式软件提供接口,它的设计与实现对嵌入式系统的整体性能有着重要的影响。嵌入式软件则为用户提供不同的功能。

随着半导体技术和计算机技术的发展,摩尔定律将会在 10 年内继续有效^[7],芯片的集成度将会进一步提高;在性能得到提升的同时,芯片的面积和价格也在不断降低。芯片集成度不断提高,使片上系统(system on chip,后文统称 SoC)技术以及片上多处理器(multi processor system on chip,后文统称 MPSoC)得到了发展与应用,以满足嵌入式系统在计算能力、通信能力和功耗方面不断增长的要求。这给嵌入式系统提供了具有更高性价比的硬件基础,嵌入式系统的应用范围进一步扩大。几乎所有的产品都可以进行芯片的集成,具备计算的能力,这种现象被称为“消失的计算机”(disappearing computer),这意味着嵌入式系统将无处不在^[8]。与此同时,应用对嵌入式系统的要求也进一步提高,嵌入式系统既要具备较强的计算能力,能够提供丰富的功能,同时也需要相应的通信能力以及较低的功耗^[9],来满足应用的需求。

1.2 SoC 的发展

随着半导体技术的不断进步,超大规模集成电路(very large-scale integrated,后文统称 VLSI)的集成密度在大幅度增加^[10]。SoC 技术是将一个系统的全部功能模块集成到单一的芯片上,从而实现在单个芯片上集成完备的系统功能^[11]。越来越多的晶体管被集成到同一芯片上,单一芯片上集成度的不断提高使得 SoC 技术得到了发展和应用。集成在 SoC 芯片上的通常是知识产权(intellectual property,后文统称 IP)核。IP 核具有可重用的特点,包括了嵌入式处理器、存储模块、接口模块和面向应用定制的处理构件等^[12]。在 SoC 上集成的 IP 核可以分为三类^[13]:软核(soft IP),是指使用寄存器传送级别(register transfer level,后文统称 RTL)或者更高级别进行描述的 IP 核;硬核(hard IP),是指具有固定的层结构,并且针对特定过程中的特定应用进行了定制的优化过的 IP 核;固化核(firm IP)是指已经做了描述但是提供了参数供设计人员进行应用定制的 IP 核。

SoC 不仅集成的晶体管数量多,而且由于集成了不同种类的功能和技术,并且实现了软硬件协同工作,使得 SoC 具有复杂的体系结构和逻辑接口^[14]。SoC 的高集成度也使得 SoC 的功能极为丰富,提高了对片上面积的有效利用,缩短了片上连线的长度,降低了系统的基础功耗,从而提高了整个系统的性能。由于 IP 核具有可重用性,在 IP 核设计完成后,相当数量的 IP 核被大多数 SoC 系统所使用和集成。在设计平台级的嵌入式系统时,这种重用性极大地提高了开发效率^[15]。

此为试读,需要完整PDF请访问: www.ertongbook.com

MPSoC 是对 SoC 技术的进一步发展,结合了 SoC 技术与多核技术的特点,是指具有多于一个嵌入式指令集处理器的 SoC^[16]。多核技术是指在同一个芯片上集成多个处理器核,以提高处理器的处理能力。如果芯片上所集成的多个处理器核相同,处理器核之间地位相同,则称为同构多核处理器;如果芯片上所集成的处理器核不同,有主处理器和协处理器之分,则称为异构多核处理器^[17-19]。MPSoC 片上既有多个处理器核,又集成了不同种类的硬件器件,具备丰富的功能,兼具 SoC 和多核的优点。

在 SoC 与 MPSoC 芯片上,往往会集成存储器,为处理器核提供存储服务,从而提高处理器的效率。

1.3 嵌入式系统中的片上存储器

随着嵌入式系统的不断发展,性能、功耗与实时性已经成为其设计的主要要求。在嵌入式系统的设计中,存储层次的设计非常重要。存储层次设计的优劣对嵌入式系统的整体性能、能耗与实现成本有极大的影响。在嵌入式系统的发展过程中,一直都是处理器速度的增长远超过动态随机存取存储器(dynamic random access memory,后文统称 DRAM)访问速度的增长。在过去的数十年中,处理器速度增长是每年 50%~100%,而 DRAM 速度的增长只有每年 7%^[20],二者间具有非常显著的增长差异。由于处理器速度增长远远超过了 DRAM 的访问速度增长,导致处理器往往需要浪费大量的时间等待较慢的存储器件的访问,以获得继续执行的数据。因此对整个系统性能影响最大的不是处理器的执行速度,而是存储器的速度。

在存储器与处理器之间一直存在着的较大的速度差距,就是“存储墙(memory wall)”问题^[21]。随着处理器计算能力的不断提高,虽然存储器的访问速度也在增加,但是由于处理器计算能力的增加速度更快,这种差距不但没有缩小,反而越来越大。处理器与存储之间的速度差距成为影响系统性能进一步提升的主要瓶颈^[22]。因此,嵌入式系统中,存储子系统一直是提高系统性能的桎梏。不仅如此,在嵌入式系统中存储子系统也是系统能耗的主要消耗点。在嵌入式系统中,存储子系统的能耗往往占到了整个系统能耗的 50%~70%^[23-24]。

SoC 技术的发展使缩小处理器与存储之间的速度差距成为可能。处理器片上不但可以集成存储器,还提供了高性能的片内总线,能够有效地提高存储器的速度,同时降低能耗。随着集成度的提升,SoC 上存储器将会占据片上面积的 50%以上^[25]。使用片上的存储器能够有效地减少系统能耗,提高整体性能^[26]。因此,很多嵌入式处理器上都集成了存储器。

在嵌入式系统中,DRAM 与 SRAM(static random access memory)是最常用的两种存储器。SRAM 的速度是 DRAM 的 10~100 倍,但是价格也是 DRAM 的

20 倍以上^[27]。因此在嵌入式系统中, DRAM 通常用作容量大的存储层次; 系统同时提供较小的 SRAM 来存储最常用的数据以减少运行时间。具有 SRAM 的系统往往会比仅使用 DRAM 的系统性能高 20% 以上。

在嵌入式系统的多级存储层次中, Cache 就是集成在片上的 SRAM。通常 Cache 是由硬件控制, 对程序员是不可见的; 而片上所集成的非 Cache 形式的 SRAM 和 DRAM 都是可编程的。由于片上集成的可编程存储器中 SRAM 较多、性能更好, 具有更典型的片上可编程存储器的特点, 本书在进行研究和测试时, 主要关注于 ScratchPad Memory。与 Cache 不同, ScratchPad Memory 是由软件控制, 即它是片上的可编程存储器^[29]。

与 Cache 相比, ScratchPad Memory 不需要众多的控制线和控制位, 所占用的片上面积更少, 因而可以在片上集成更大容量的存储器; ScratchPad Memory 在设计上比 Cache 简单, 访问通过寻址实现, 访问 ScratchPad Memory 所需的能耗比 Cache 小; ScratchPad Memory 具有可编程的特点, 这使得能够更好的通过程序和系统优化对 ScratchPad Memory 进行控制。因此, ScratchPad Memory 在嵌入式系统中得到了越来越广泛的使用。

1.4 片上软件系统的研究意义

嵌入式系统的发展迅速, 其应用也越来越广泛。嵌入式系统以应用为中心, 其软、硬件的设计与实现都是围绕具体的应用环境进行定制, 也必须依赖于提高使用者的体验。随着嵌入式系统应用的不断深入, 性能、实时性与能耗等方面对于嵌入式系统的要求越来越高, 并且这些要求有可能相互之间是冲突关系^[30]。嵌入式系统的软、硬件需要协同进行, 对系统进行高效率的设计, 以满足应用的需求。基于 SoC 与 MPSoC 的嵌入式系统研究的不断深入, 片上面积的利用效率越来越高, 所集成的器件种类丰富, 性能和容量也不断提升。利用片上高性能器件来提升嵌入式系统的整体性能更是当前嵌入式系统研究的重点之一。由于片上可利用面积的增加, 越来越多的嵌入式处理器将存储器集成在片上, 通过片上高速总线来进行访问。片上存储器的集成度提高, 容量和速度也都大为增加。片上存储器有利于缩小处理器核与内存之间的访问延迟, 提高系统的整体性能。

目前最为常用的片上存储器类型是 Cache。Cache 一直是桌面处理器的标准存储器。Cache 的主要优点是它由硬件控制, 对它的各种操作由硬件自动完成。因而 Cache 能够对常用数据进行自动管理, 进而提高系统性能。Cache 对于系统中运行的程序来说是透明的。硬件提供对 Cache 操作的支持, 包括读写 Cache、从内存中取数据、向内存中写入数据, 以及对 Cache 数据的管理^[31]。

然而在嵌入式系统中, 由于运行在处理器上的程序是受硬件资源条件限制的, 因此 Cache 的性能远不如桌面处理器的 Cache。研究表明, 将 Cache 用于嵌入式系

统时,所消耗的能耗更高^[32],占用了更多的片上面积,并且性能上的表现远远不如在桌面处理器上的表现^[33-34]。因此尽管 Cache 在嵌入式系统也得到了应用,但是由于 Cache 会降低最坏情况时的性能,是否在嵌入式系统中采用 Cache 结构是有争议的^[28]。

由于 Cache 的种种不足,出现了另一种将 SRAM 集成到片上的方式,即 ScratchPad Memory 的方式。ScratchPad Memory 是集成到芯片上的非 Cache 用途的 SRAM 存储器的统称,也是片上可编程存储器的主要形式。在某些高端嵌入式处理器上,可以通过特定的开关在 ScratchPad Memory 和 Cache 之间进行转换,这使 ScratchPad Memory 具有了更大的优势。通过软件控制的方法,将数据存放在 ScratchPad Memory 中,提供给处理器使用。由于 ScratchPad Memory 具有软件可控制、响应速度快、占用片上面积小、能耗低的特点,在嵌入式系统中得到了越来越广泛的应用。

现有的研究主要是对单个的嵌入式应用程序进行优化,其主要方式是通过通过对单个程序进行分析,选取适当的程序片段,通过经过控制的编译过程,将选定的程序片段分配到 ScratchPad Memory 上。研究表明,通过 ScratchPad Memory 进行优化,能够有效地提高程序运行的速度,减少程序运行时的能耗。但是由于只对单一的应用程序进行分析和优化,现有的研究缺少对多程序并行的优化,也缺少对嵌入式操作系统的优化研究。因此,现有的研究具有较大的局限性,尽管能够在一定程度上提高系统性能,但是对 ScratchPad Memory 的利用率不高,并实际使用的可行性较少。

本书关注以 ScratchPad Memory 为代表的片上可编程存储器的有效使用,研究基于片上可编程存储器尤其是 ScratchPad Memory 的嵌入式系统优化,提出了片上软件系统的设计思想,通过对嵌入式操作系统进行优化、多道程序共享 ScratchPad Memory 的优化以及对 MPSoC 进行优化,来提高嵌入式系统的整体性能,形成以片上可编程存储器为基础的嵌入式软件系统整体解决方案。

第 2 章 片上可编程存储器

片上软件系统的研究以片上可编程存储器为基础。本章首先介绍 Cache 作为嵌入式系统的片上存储器的研究,分析 Cache 用于嵌入式系统的缺点。随后对既有的对片上可编程存储器的研究进行分析,对各种不同的片上可编程存储器优化技术做分类综述。最后在对优化方法进行分析 and 总结的基础上,提出现有方法所存在的问题的解决方法。

2.1 片上可编程存储器概述

2.1.1 片上可编程存储器的特征

片上可编程存储器是指可进行编程的片上存储器,是对程序员可见的,程序员可以像使用内存一样使用片上存储器。片上可编程存储器分为两类:一类是集成到片上的非 Cache 形式的 SRAM(即 ScratchPad Memory);另一类是集成到片上的 DRAM。DRAM 是由一个晶体管构成的,为了防止信息的丢失,使用 DRAM 时必须周期性的对 DRAM 进行刷新;而 SRAM 则是由 6 个晶体管构成^[25],在使用 SRAM 的过程中,不需要定时对电路进行刷新。因此 SRAM 在存储过程中所消耗的电能比较少,而读写的性能比较高;但是由于 SRAM 追求的是高速和稳定性,在集成度上不如 DRAM,而且成本相对较高。在进行片上可编程存储器的研究时,一般主要关注于 ScratchPad Memory 的研究,其方法同样可用于以 DRAM 为硬件基础的片上可编程存储器。本书后续章节中所述片上可编程存储器是指 Scratch-Pad Memory。

由于 SRAM 的上述特点,这种存储器首先被用作计算机系统中的高速缓存即 Cache。Cache 很早就被提出用做处理器与内存之间的数据缓冲,并且在通用计算机中,Cache 被证明能够有效地提高系统的性能。很多嵌入式处理器尤其是高端的嵌入式处理器也集成 Cache,以提高对存储器的读写速度。与内存相比,Cache 具有更快的访问速度,有利于减少存储访问延迟。在嵌入式系统中,SRAM 同样也被集成到片上作为 Cache 使用。

Cache 是完全使用硬件进行管理和调度的,因此它对于程序员来说是不可见的,程序员难以在程序设计阶段来使用它。基本的 Cache 结构如图 2.1 所示^[35]。Cache 一般来说会被划分成不同的块,与内存中的存储块进行映射。Cache 的每个单元存储块需要增加相关标志来确认数据是否装入 Cache 以及装入 Cache 的数据是否有效。其中标志位表示对应的内存块是否已经装入 Cache 当中,有效位用来标记装入 Cache 中的数据是否有效,以说明内存中对应块在 Cache 中的副本是否正确有效。

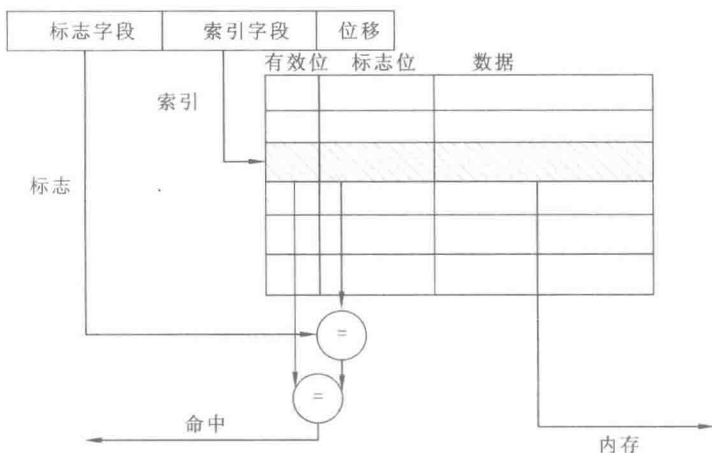


图 2.1 Cache 基本结构示意图

当处理器需要进行数据读写操作时,首先会对 Cache 进行读写。在对数据进行操作之前,需要通过地址当中的标志字段和索引字段分别与 Cache 的标志位和有效位进行比较,确认所请求的数据是否已经装入 Cache 中并且是否是有效的数据。如果所请求的数据在 Cache 中并且是有效数据,将对数据进行操作,这就是 Cache 命中;否则就称为 Cache 失配,这是就需要访问更低层的存储器,将数据从内存搬运进 Cache。此后再访问同一数据就不用再从内存中获取。

将 Cache 应用在嵌入式系统中能够在一定程度上提高系统的性能,但是与片上可编程存储器相比,尽管两者都是由 SRAM 集成到片上,由于运行机制不同,它也会存在以下一些问题。

第一,在嵌入式系统的应用中,实时性是必然要求,必须考虑实时任务的处理。尽管 Cache 有利于降低存储访问延迟,但是由于 Cache 失配时会产生长时间的延迟,这种延迟的长度也是不可预测的,这使得在使用 Cache 后访问时间出现了不确定性^[36],对系统性能具有重大的影响。与此同时,Cache 可能会出现冲突,从而给存储系统增加额外的负担并影响正在进行中的存储访问,这将降低系统的性能。片上可编程存储器是直接访问,不存在这一问题。

第二,Cache 需要更多的片上面积。Cache 由于是硬件控制,需要在设计时增加连线,在对 Cache 进行读写操作时,又需要额外的标志位、有效位和比较器进行数据装入与否和有效性的判断,这大大增加了 Cache 所占用的片上面积,减少了可

以有效利用的空间^[37],同时,这也导致了成本的增加。片上可编程存储器是软件编程控制,不存在 Cache 那样的比较过程,不需要额外的标志位、有效位和比较器,减少了占用的片上面积和能耗。

第三,嵌入式系统往往是采用电池供电^[38],对功耗和电池的寿命特别关注。Cache 访问的复杂性和较大的片上面积使得它的能耗较大,不利于满足嵌入式系统对能耗的要求。而片上可编程存储器通过地址访问,不需要额外的控制线和控制位,减少了访问的复杂度,也减少了所需要的片上面积,从而能够有效地降低系统的能耗。

针对片上可编程存储器与 Cache 的研究^[39]结果表明,片上可编程存储器的能耗比 Cache 少 40%。由于减少了用作标记位等的容量和连线的数量,所占用的片上面积是 Cache 的 66%。同时,与 Cache 相比,使用片上可编程存储器作为片上存储器,能够将程序的性能提升 18%。此外,由于片上可编程存储器可由程序员或者程序控制使用,程序运行时对存储的访问是可预期的,有利于系统实时性的保证。因此无论从片上面积、实时性、成本、功耗等各个方面来说,片上可编程存储器都远远地超出了 Cache,这使得片上可编程存储器在嵌入式系统中将逐渐取代 Cache,占据片上存储器的主要地位^[39-40]。

2.1.2 集成片上可编程存储器的单核处理器结构

集成于单核处理器上片上可编程存储器通常与 Cache 并存于片上。典型的集成了片上可编程存储器的单核嵌入式处理器的框架图如图 2.2 所示^[29]。处理器核通过片上总线与片上可编程存储器、Cache 和外部存储器接口(external memory interface)相连;外部存储器接口则负责处理器核、Cache 和片上可编程存储器与外部存储器的连接。

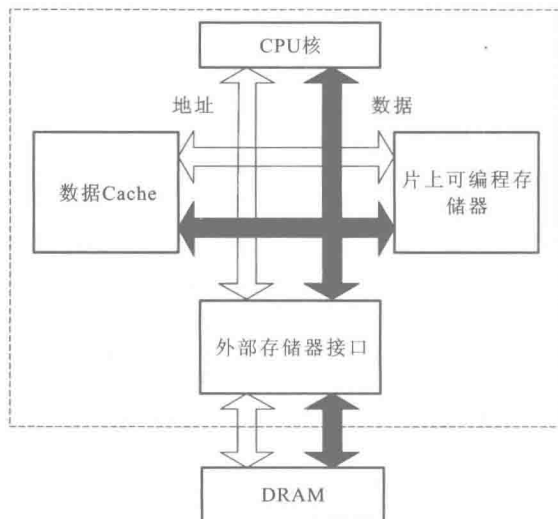


图 2.2 集成片上可编程存储器的单核处理器结构示意图