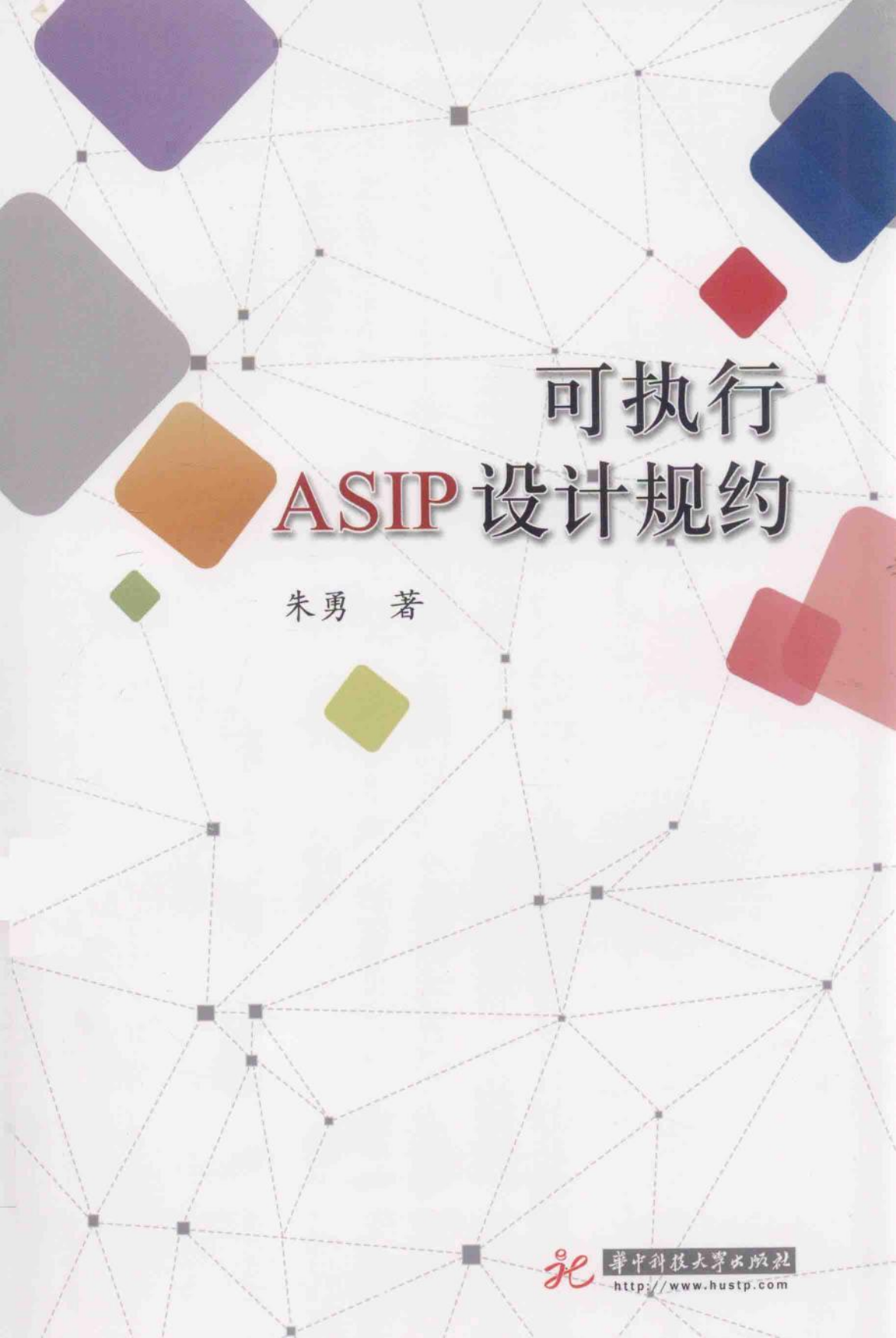




可执行 ASIP 设计规约

朱勇 著



华中科技大学出版社
<http://www.hustp.com>

可执行 ASIP 设计规约

朱 勇 著

华中科技大学出版社
中国·武汉

内 容 简 介

本书系统地论述了 ASIP(专用指令集处理器)设计规约及系统设计方法学。首先,从一般系统层面介绍基本的开发流程,阐述“描述-综合”设计方法学的概念,以此作为 ASIP 设计的基本原则。接下来,第 4 章给出系统的基本模型及处理器体系结构,作为 ASIP 建模参考。第 5 章详细列出各类体系结构描述语言(ADL)的特征,以作为 ASIP 描述手段。第 6 章指出 ASIP 设计规约的内容。在“描述-综合”设计方法学原则指导下,采用 ADL 描述手段,得到指令集和结构的逻辑综合。最终,给出原型机运行仿真结果及集成环境主界面,并总结出 ASIP 设计的重要原则。

本专著可作为研究计算机体系结构、SoC(片上系统)和系统设计方法学相关学者的参考书,也可以供计算机、电子信息和系统科学等专业的研究生研读。

图书在版编目(CIP)数据

可执行 ASIP 设计规约/朱勇 著. —武汉:华中科技大学出版社,2013.12

ISBN 978-7-5609-9528-1

I. ①可… II. ①朱… III. ①集成电路-芯片-设计 IV. ①TN402

中国版本图书馆 CIP 数据核字(2013)第 286999 号

可执行 ASIP 设计规约

朱 勇 著

策划编辑:谢燕群

责任编辑:熊 慧

封面设计:朱 翥

责任校对:周 娟

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:武汉市洪山区佳年华文印部

印 刷:武汉科源印刷设计有限公司

开 本:710mm×1000mm 1/16

印 张:10.5

字 数:212 千字

版 次:2013 年 12 月第 1 版第 1 次印刷

定 价:28.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

作为一名长期从事嵌入式处理器应用项目(包括单片机测试仪、工控机系统、嵌入式处理器接口和 IP 核处理单元等),以及计算机系统结构的研究者,一直致力于 CPU 核心机理研究,此次有幸得到国家自然科学基金“ASIP 行为逻辑及其综合的关键技术研究”(项目编号:61142006)的资助,对 ASIP(专用指令集处理器)设计方法学进行了专门研究。

本研究采用 ADL(体系结构描述语言)描述方法,研究如何用规范、机器可读形式建立 ASIP 体系结构模型方法学,即可执行规约。研究内容包括:各视图的 ADL 描述,如指令集 ADL 描述、流水线微结构 ADL 描述、流水线 Petri 网 PNML(Petri 网标记语言)描述等,构建指令集行为模型和流水线微结构模型。将上述 ADL 描述在 EDA 环境中进行仿真评估,加强模型验证与优化。课题主要解决以下两个关键问题:① ADL 能够捕捉多种体系结构,包括 RISC、VLIW、TTA 等。② 构建流水线 PNML 模型。欲达到的目标是,建立基于 ADL 描述的 ASIP 流水线 Petri 网模型,经 EDA 工具捕捉、优化后,自动综合为逻辑实现,使可执行设计规约驱动整个设计流程,从而提高设计效率。

ASIP 设计在计算性能和灵活性的能耗方面获得很好平衡。其特色是具有用户定义的指令、数据路径和加速器。与 ASIC(专用集成电路)和 GPP(通用可编程处理器)相比,ASIP 应用实现了二者间满意的性价比平衡,具有良好的应用前景。其主要优点是最大限度地利用有限的硬件资源,且这一技术特点使得其越来越受到嵌入式系统设计人员的青睐。

因处理器在信息领域中的重要地位,其核心技术一直是科技发达国家的研究重点,特别是一些超级公司更是垄断了 GPP 产业链。反观国内厂商的产品,这些产品大多是采用国外主流 SoC 平台的衍生品,基于自主知识产权的 ASIP 核心寥寥无几。与 GPP 和 ASIC 非常成熟的研究相比,ASIP 系统设计目前还缺乏通用的系统设计流程和开发工具,设计上更多地依靠系统设计人员的经验,很多问题还有待进一步研究。其中 ASIP 系统设计方法学是亟待解决的问题。

相对于成熟的 GPP 设计,ASIP 体系结构 DSE(设计空间探索)具有更大的空间。特别是新型的可配置及 TTA 架构的 ASIP 模型验证、仿真及其逻辑实现,极大地扩展了计算机体系结构研究领域。借助 SoC 硬件技术,动态数据路径及用户定义指令在满足功耗和面积约束下,能够解决嵌入式应用关键功能瓶颈。基于

Petri 网的流水线模型可以更好地实现状态迁移的并行性与同步,通过编码树将处理器指令行为分解到不同的流水段,并映射到相应的功能单元,提高流水线效率。

有些 ADL 在国外已成功运用于 ASIP 设计,其最显著的特征是捕捉处理器体系结构并生成工具链。作为处理器精确建模规约,它可用于设计自动化的各项任务,如仿真、编译、综合、测试、体系结构验证等;作为设计语言,它借鉴编程语言特性与功能,特别是先进的软件工程框架实现可执行设计规约。ADL 不仅专注于处理器及存储器体系结构,还具备描述 SoC 外围设备与接口的能力,能够进一步捕捉多核体系结构,并支持 RTOS。

总之,ASIP 设计方法学涉及计算机体系结构、设计方法学、模型规约及软件工程框架等方面的研究,具有重要的科学研究意义。

本课题的一个重要成果是开发了具有自主知识产权的集成设计环境(已获得两项软件知识产权),为 ASIP 完整工具链的建立奠定了基础。

国内罕有 ASIP 设计方法学方面的专著,《可执行 ASIP 设计规约》算是对此的一个补充。

最后要感谢对本专著给予支持的单位:国家自然科学基金委员会和华中科技大学出版社。

将此书献给我至亲的家人!

朱 勇

2013 年 11 月于武昌

目 录

第 1 章 引言	(1)
1.1 嵌入式系统与片上系统	(1)
1.1.1 嵌入式系统	(1)
1.1.2 IP 核设计与复用	(3)
1.1.3 软硬件协同设计	(4)
1.2 专用指令集处理器	(6)
1.2.1 ASIP 特性	(6)
1.2.2 研究状况	(7)
1.3 设计方法学	(9)
1.3.1 体系结构描述语言	(9)
1.3.2 描述综合设计方法学	(11)
1.3.3 可执行设计规约	(12)
第 2 章 系统开发	(14)
2.1 基本概念	(14)
2.1.1 分析	(14)
2.1.2 设计	(17)
2.1.3 生命周期	(20)
2.1.4 测试	(21)
2.2 系统分析	(22)
2.2.1 需求分析	(22)
2.2.2 结构对象分析	(23)
2.2.3 行为对象分析	(25)
2.3 系统设计	(27)
2.3.1 架构设计	(27)
2.3.2 机制设计	(28)
2.3.3 详细设计	(29)
第 3 章 设计方法学	(31)
3.1 设计方法学概述	(31)
3.1.1 惯例设计方法学	(31)

3.1.2	系统级设计方法学	(33)
3.1.3	MDA 设计方法学	(35)
3.2	系统描述	(36)
3.2.1	设计表示	(36)
3.2.2	UML 视图	(37)
3.2.3	Y-chart 模型	(38)
3.2.4	抽象级别	(40)
3.3	设计综合	(41)
3.3.1	“描述-综合”方法学	(41)
3.3.2	系统综合	(42)
3.3.3	ASIP 综合	(44)
3.3.4	逻辑与时序综合	(45)
第 4 章	模型与体系结构	(48)
4.1	典型模型	(48)
4.1.1	模型类型	(48)
4.1.2	面向状态的模型	(49)
4.1.3	面向活动的模型	(51)
4.1.4	面向结构的模型	(52)
4.1.5	面向数据的模型	(53)
4.1.6	异构模型	(54)
4.2	体系结构	(57)
4.2.1	体系结构分类	(57)
4.2.2	通用处理器	(57)
4.2.3	并行处理器	(60)
4.2.4	专用体系结构	(61)
4.3	微结构	(62)
4.3.1	概念	(62)
4.3.2	实例	(64)
第 5 章	体系结构描述语言	(67)
5.1	UML 建模语言	(67)
5.1.1	UML 基础	(67)
5.1.2	嵌入式领域 UML 描述	(69)
5.2	Spec 描述语言	(71)
5.2.1	SpecCharts 语言	(71)

5.2.2	SpecC 语言	(74)
5.3	LISA 指令描述语言	(76)
5.3.1	LISA	(76)
5.3.2	LPDP	(78)
5.4	HDL 硬件描述语言	(82)
5.4.1	VHDL	(82)
5.4.2	Verilog	(83)
5.4.3	UML 至 HDL 的映射	(85)
第 6 章	ASIP 系统规约	(88)
6.1	开发模式	(88)
6.1.1	模型规约	(88)
6.1.2	可执行模型	(89)
6.1.3	开发流程	(91)
6.2	建模模式	(92)
6.2.1	对象模型	(92)
6.2.2	行为模式	(93)
6.2.3	调试与测试	(96)
6.3	框架设计	(97)
6.3.1	概述	(97)
6.3.2	实时框架	(100)
6.3.3	可执行框架	(102)
6.3.4	设计原理与度量	(104)
第 7 章	ASIP ADL 描述	(108)
7.1	ASIP 模型	(108)
7.1.1	LISA ADL 描述	(108)
7.1.2	Expression ADL 描述	(111)
7.1.3	模型抽象	(116)
7.2	TTA	(117)
7.2.1	OTA 与 TTA	(117)
7.2.2	指令集行为	(118)
7.2.3	ADF 规约	(119)
7.3	流水线微结构	(123)
7.3.1	流水线状态机	(123)
7.3.2	Petri 网模型	(124)

7.3.3	PNML 描述	(126)
第 8 章	ASIP 综合实现	(129)
8.1	ADL 综合	(129)
8.1.1	综合驱动探索	(129)
8.1.2	ISA 综合	(131)
8.1.3	结构综合	(134)
8.2	软件框架	(138)
8.2.1	MVC 架构	(138)
8.2.2	ThinkPHP 框架	(140)
8.2.3	应用软件编程	(142)
8.3	集成环境	(145)
8.3.1	环境形态	(145)
8.3.2	时空设计模式	(147)
8.3.3	IDE 组成	(148)
第 9 章	结束语	(150)
9.1	结果	(150)
9.2	结论	(153)
9.3	前瞻	(157)
参考文献		(158)

第 1 章 引 言

计算机体系结构按其应用,分为服务器、桌面机和嵌入式系统三个领域。近几年,无论是增长速度还是覆盖范围,嵌入式应用领域都备受人们关注。ASIP(application specific instruction processor,专用指令集处理器)已成为嵌入式处理器的主要选择,必将成为下一代信息处理的重要手段。

嵌入式系统从初期裁剪通用计算机架构的泛嵌入式形态发展为针对特殊要求而采取的优化关键路径及定制用户指令的专业生态环境。后者体现在以下三个方面:① 硬件采用基于 IP 核复用与集成技术的 SoC(system-on-chip,芯片级系统)芯片,通过软硬件协同设计整合嵌入式系统;② 处理器采用 ASIP,填补了专门硬件 ASIC(专用集成电路)与可编程 DSP(数字信号处理器)针对应用领域间的“能耗-灵活性”空间,具有优化目标应用自定义指令及数据路径的特性;③ 基于 ADL 的“描述-综合”设计方法学,捕捉 ASIP 的“功能-结构-物理”模型,为可执行规约设计奠定了原生基础。

1.1 嵌入式系统与片上系统

1.1.1 嵌入式系统

Hennessy 和 Patterson 将计算机系统体系结构分为桌面机、服务器和嵌入式系统三大类^[1]。桌面机追求性价比,加强程序执行性能,不太注重程序大小和处理器的功耗;服务器强调可扩展性,即客户对数据库、文件系统和网络应用的多用户服务请求;与它们不同,嵌入式系统则强调实时、成本和功耗问题。

随着应用对嵌入式系统的要求越来越高,处理器结构不断发展,嵌入式系统也越来越复杂,8 位和 16 位处理器已经不能胜任复杂应用系统的需求,32 位嵌入式系统处理器应运而生。许多嵌入式处理器结构是从过去的主要用于桌面系统的结构中衍生出来的,与桌面处理器之间的不同也不再仅仅局限于寄存器组织、基本指令集、流水线结构,而是扩展到了诸如功耗、成本、外设集成度、中断响应时间、片上 RAM/ROM 的容量等诸多问题上。

桌面系统处理器的典型代表是 Intel x86 系列处理器,伴随着非常丰富的软件资源环境,占据了桌面系统市场很大的份额。在嵌入式系统领域,目前采用主流结

构的代表是 ARM 系列处理器,它有众多的体系结构系列及越来越普及的开源软件支持,使得其应用受众不断增加。同时,其他诸如 MIPS 系列处理器、Motorola 的 68K 系列处理器及单片机控制器在工业控制和通信系统中仍有较大优势。

嵌入式处理器大多采用 RISC(精简指令集计算机)体系结构。RISC 的概念最初由 Patterson 和 Ditzel 在其 1980 年发表的论文中提出来。由于当时 RISC 在同样的技术条件下可以将计算性能提高 2 到 5 倍,因此 RISC 的概念一提出便引起了业界极大的兴趣。最初的商用 RISC 处理器主要有 3 种:Berkeley 的 RISC I、RISC II 处理器,IBM 的 801 处理器和 Stanford 的 MIPS 处理器。它们都采用 load/store 结构,并强调有效的流水线;其指令结构较为简单,是固定形式的 32 位指令字,一般一条指令只能执行一个操作,从而实现了处理器的高时钟频率和高性能表现。RISC 坚持两个主要设计原则:① 将指令严格地区分为 load/store 类访存指令和其他 R-R 类指令;② 尽量地利用简单指令。

嵌入式系统有两个设计目标:一是尽量减少产品成本及缩短上市时间;二是满足系统性能与功耗要求。也就是说,嵌入式系统设计的最终目的是找到产品成本、性能和功耗之间的相对平衡点,所以在设计过程中需要综合考虑以下因素。

(1) 使硬件结构适应产品的功能,以便能够针对特定的应用选择成本最小的解决方案。

(2) 在新的应用中尽可能地重复使用已经存在的成熟设计,减少重复设计,即从一个更大的“库”中直接进行选择,拼装成最终的产品。

(3) 尽量考虑多个应用都能够使用的设计,提高产品的涵盖范围。

(4) 调整系统性能,使之满足应用需求,例如,对于低端应用,不需要设计高性能的多媒体和通信处理算法。

(5) 注重低功耗设计。

根据这些设计考虑及嵌入式应用的发展趋势,下一代嵌入式系统必将会朝着片上系统的方向发展。20 世纪 90 年代中期,更多的 32 位处理器推向市场,开始出现多媒体扩展结构。同时,RISC 与 DSP 的融合结构将目标瞄准高端嵌入式处理器市场。20 世纪 90 年代末期,芯片制造工艺更加先进,集成度越来越高,不同用途的处理器核可以集成在一块芯片中实现,使得整个嵌入式系统进入片上系统发展时代。随着 SoC 研究的逐步深入,要求嵌入式系统具有更强的整合性,更注重软硬件的协同设计,对多媒体处理能力和功耗问题提出更高的要求。嵌入式系统处理器也向着多元化的趋势发展,不同的体系结构间相互融合,取长补短。

SoC 是以 IP 核复用为基础,以软硬件协同设计为主要设计方法的芯片设计技术。其关键是利用经过验证的 IP 核,并成功地把 IP 核集成到 SoC 系统中。

1.1.2 IP 核设计与复用

IP 核有软核(soft IP)和硬核(hard IP)两种类型。前者以可综合的硬件描述语言(HDL)代码的形式交付;后者则用制定的工艺进行功耗、面积或者性能的优化,以 GDS II 格式交付。成功地开发 SoC IP 核结构,需要做到以下几点^[2]。

(1) 硬化:优化配置并使软核硬化。

IP 核硬化过程就是在标准规定的速度、功率和范围内以目标工艺实现 IP 核。该实现必须能够提供准确的建模、自动化方法,工艺易于移植,以及具有基于业界标准的电子设计自动化(EDA)工具。硬化过程首先需要 IP 核供应商提供高质量 RTL(寄存器传输级)描述,并且提供一套完整的 GDS II 设计实现方案。

(2) 建模:高度精确地为硬化的软核自动建模。

典型的 SoC 设计流程包括功能模型设计、时序模型设计、测试模型设计、物理模型设计和功率模型设计。

功能模型必须代表系统仿真中的 IP 核周期特征,并且能够在门级仿真中支持精确到比特级的 RTL 仿真和时序注释。此外还应消除仿真器的特殊结构和接口,在仿真环境中便于移植。由于模型构建过程实际上是 C/HDL 设计的二进制目标模型的构建过程,考虑到最终用户设计环境的不可测性,所有通过使用编译器构建的模型必须能够在仿真器,各种语言、硬件平台和操作系统上充分移植。该模型的功能性和时序分开,因此只要提供正确的级别时序信息,就能在各种设计的抽象级别使用。

时序模型具备所有的时序特点,在逻辑综合、物理综合、测试综合、静态时序分析、时间驱动地点和路线等设计步骤中得到采用。因此,时序模型的准确性就变得尤为重要,必须达到以下要求:黑匣子、独立性、准确性及符合业界标准。

IEEE(美国电气和电子工程师协会)规定了一个 IP 核测试语言(CTL),定义了嵌入式 IP 核和 SoC 的测试接口,为 SoC 互联和逻辑提供了可测试性。CTL 支持多种测试方式,如 Scan(扫描)和 BIST(内建自测试)等。构建测试 IP 核的程序包括:对 IP 核进行测试包装;在包装后的内核上执行多模型 DRC(设计规则检测);为包装后的 IP 核指明 CTL 种类;为包装后的 IP 核构建测试模型。

物理模型是 IP 核具体物理实现的抽象,必须包含足够的信息满足以下设计实现的需要:平面布置、置换和具体线路安排。该模型必须准确表述:元件占用面积(IP 核本身的面积,即长和宽数据)、接口引脚/端口数量(每个 IP 核 I/O 端口的名称、尺寸、形状、位置、连接层和相应引脚的数量)、线路障碍(IP 核上线路障碍的面积、形状和影响层面等)、电源和接地(每个电源连接的名称、面积、形状、位置和接线层)。

功率模型描述了 IP 核功耗,必须忠实反映静态和动态功耗、I/O 端口和内部节点的开关状态、I/O 端口和内部节点的状态、运行方式、电压和温度等条件、电容负载和输入瞬变时间。

(3) 集成:将模型综合到 SoC 设计流程中。

选择 IP 核时首要考虑的因素是,IP 核与目标系统的配合程度。为了使开发的 IP 核能够高效地集成到新的设计中去,设计复用(reuse)和标准化是必由之路。IP 核集成必须解决的重要问题有:在系统结构设计和模块划分时,选择合适的片上总线结构和 IP 核库;模块间的接口协议尽可能简单,接口定义尽可能与国际上通用的接口协议一致;慎重考虑时钟分布,以及电源、地线的走线,要投入较大精力对关键路径进行优化;注重积累集成的经验。对于成功集成的 IP 核,应该进一步完善,同时记录下来,形成技术文档。

(4) 验证:IP 核是否符合设计者的想法。

IP 核验证技术包括以下几类。

① 目的性验证:其目的是验证设计者所预想的功能是否在设计中得到正确实现。通常,它在最高抽象层次上完成。其最终结果是建立一个所谓的“黄金模型”。该模型可以作为整个设计过程中各种更加详细的设计视图的参考基准。

② 等效性验证:其目的是验证在设计过程中生成的不同层次的设计功能是否与“黄金模型”功能相一致。

③ IP 核验证:IP 核验证是指对单个 IP 核的功能进行验证的过程,即单元测试。

④ 集成验证:集成验证是指对包含一个或多个 IP 核的 SoC 进行功能验证的过程,即 SoC 的系统级验证。其重点是 IP 核的连接和相互作用,验证所用模型是否能精确地仿效 IP 核接口。

可供复用的 IP 核既要有说明核的功能时序要求等的功能描述文件,还要有设计实现和验证两个方面的文件,即还应包含实现 IP 核的 RTL 源代码或网表文件,以及用于 IP 核实现后逻辑验证的仿真模型和测试向量。其设计流程主要包括设计规约与划分、子模块的规范与设计、测试台开发、时序检测和集成等。在选择 IP 核时要考虑多方面的因素,如与目标系统的配合程度、成本、上市的紧迫性、与其他模块的兼容性、系统功能、IP 核的质量及其支持性(标准的遵从程度、未来发展的蓝图、易用性、获取 IP 核授权的效率、合作厂商的可依赖程度)、集成的难度等。

1.1.3 软硬件协同设计

软硬件协同设计不仅是一种设计技术,同时也是一种新的设计方法学,其核心

问题是在设计过程中协调软件子系统和硬件子系统^[3]。与传统的嵌入式系统设计方法不同,软硬件协同设计强调软件和硬件设计开发的并行性和相互反馈,克服了传统方法中把软件和硬件分开设计所带来的种种弊端,协调软件和硬件之间的制约关系,达到系统高效工作的目的。它提高了设计抽象的层次,拓展了设计覆盖的范围,与此同时还强调利用现有资源,即重用构件和 IP 核,缩短系统开发周期,降低系统成本,提高系统性能,保证系统开发质量。嵌入式系统软硬件协同设计方法主要包括系统建模、协同综合、协同仿真、协同验证、设计功能和性能指标评价、SoC 测试调度等方面,并且还分为不同的设计层次,相关技术简述如下。

(1) 协同综合技术:嵌入式系统是软硬件一体化的系统,其中功能既可以由硬件来完成,也可以由软件来实现。硬件速度快,而软件成本低,这就需要权衡系统的时间、成本等性能指标之间的关系。决定系统各个模块由硬件完成还是由软件实现,这是一项非常重要的工作。这个划分系统软硬件的过程称为协同综合或者协同划分。

(2) 协同验证技术:在系统设计的各个阶段,根据系统性能指标要求对设计方案综合评价,以验证系统的合理性和可行性。协同验证的研究主要有两个方向,即仿真验证和形式化验证。仿真验证方法用硬件描述语言 HDL 完成对硬件子系统的描述,完成系统软硬件的联合调试,纠正其中的设计错误;形式化验证方法建立被验证系统的数学模型,然后用数学方法证明被验证系统的正确性,以及各种性能指标是否满足要求。

(3) 性能指标评价技术:嵌入式系统是以应用为中心,对系统功能和性能指标(成本、面积、功耗、实时性)都有严格要求的专用计算机系统。对各性能指标进行标准评价是保证嵌入式系统需求的必要条件。

(4) SoC 测试调度问题:随着设计复杂度的增加,SoC 上重用的 IP 核数量越来越多,为了缩短芯片测试的时间,需要尽可能并行地测试芯片上的 IP 核。将 IP 核的输入/输出端动态分配到同一测试总线上,允许它们同时进行测试,即在设计中进行测试调度。

SoC 是将若干处理器单元和外设集成在一起的一个系统组成,可以为应用提供高集成度的芯片解决方案。将系统调度与控制、数字信号处理等功能集成在一块芯片中实现是嵌入式系统发展的必然趋势。对于集成了控制调度处理器、数字信号处理器、专用电路处理器等多种结构的 SoC 设计,其设计方法与设计工具越来越受到芯片设计方的关注。设计 SoC 系统时,不仅需要进行软件和硬件的协同设计,还需要一个强大的系统设计平台给予支持。在开发平台中,需要进行体系结构和微结构的划分与设计,进行功能的确定和系统应用的开发,对各个设计层次和功能进行匹配与比较,最终达到一致。这一匹配、比较过程是在系统设计过程中分

阶段、分步骤进行验证、仿真、综合、测试的过程。允许硬件重用的一组结构称为硬件平台,它必须在一定的层次上进行抽象,使得应用软件能够看到硬件的更高层次接口,即应用程序接口 API。除此之外,还需要有一个软件层次来执行这一抽象,并对硬件平台中的不同部分进行约束,如通过实时操作系统对处理器核和存储系统进行操作,通过设备驱动程序对 I/O 系统进行存取,通过网络通信系统对网络接口进行控制等等,这一层次称为软件平台。在过去 10 年中,集成数字电路作为运算核心有了很大发展,然而其设计效率的增长仍然赶不上技术的发展速度,其间的差距表明具有竞争性的新型设备实现出现了严重瓶颈。针对此问题,使用嵌入式数字设备是实现硬件向软件转换的一个解决方案。得益于实现过程的简化及重用应用因素的增加,软件设计的效率在不断增强。基于平台环境的软硬件协同设计、验证和仿真能够适应嵌入式领域的快速发展和 SoC 设计时代的要求。

1.2 专用指令集处理器

1.2.1 ASIP 特性

与面向桌面应用的通用处理器相比,嵌入式处理器提供了卓越的性价比,不过后者在计算性能上仍有一定的局限性。由于面向广泛的嵌入式应用,因此这些处理器在某些专门应用上显现出较差的能耗效率。

从图 1-1 可以看出,ASIP 能够填补专门硬件 ASIC 与可编程 DSP 针对应用领域间的“能耗-灵活性”空间^[4]。它具有为优化某一目标应用而自定义指令及数据路径的特性,其结果是,具有比通用处理器更优的计算性能和能耗表现。ASIP 的主要优点是,最大限度地利用有限的硬件资源。这一技术特点使其越来越受到嵌入式系统设计人员的青睐。这也是当今业界越来越多地使用定制处理器的原因之一。

从硬件设计师的观点来看,通过跟踪标准的改进,ASIP 促进了具有高级灵活性任务的实现,特别是基于软件实现的高可重用特性使得设计时间大大缩短,对于实现已有产品显著特征以使其具有竞争性目标有特别意义。并且,ASIP 可以用高级语言建模,以快速与系统方法共享硬件设计资源。可综合的 ASIP 是一种独立技术,能够与其他硬件模块通过半定制设计流程集成在一起。

从软件设计师的观点来看,ASIP 提供了一个新的优化自由度。其设计输入为高级语言的软件实现和硬件描述语言描述的体系结构。通过可扩展处理器资源,它还为软件设计师和硬件架构设计者去除了常规固定核处理器的计算性能障碍。因此,用高能耗效率的 ASIP 替代传统固定核处理器可以满足嵌入式应用的

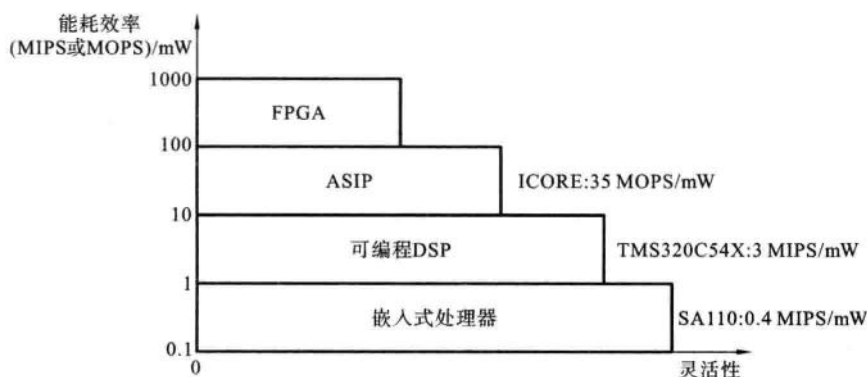


图 1-1 “能耗-灵活性”空间

性能约束。

ASIP 设计是一个复杂的优化问题,它涉及大规模集成电路逻辑、计算机体系结构和应用软件设计三方面内容。设计任务的复杂性让设计者在有限的时间内探索大量的备选方案以获得最优实现非常困难,并且,对系统能耗效率严苛的要求增加了额外的复杂性,加大了系统设计的难度。

1.2.2 研究状况

国内外研究 ASIP 设计方法学的主要机构和团队有如下一些。

(1) 亚琛工业大学通信技术和嵌入式系统研究所(the Institute for Communication Technologies and Embedded Systems, ICE)由 Gerd Ascheid 教授领导的集成信号处理系统(ISS)研究室、Rainer Leupers 教授领导的硅系统软件(SSS)研究室和 Anupam Chattopadhyay 教授领导的多处理器体系结构研究室组成,从事数字通信系统传送与接收算法、嵌入式处理器体系结构和电子系统级设计工具方面的研究。

(2) 坦佩雷理工大学(TUT)计算机系 ASIP 灵活设计方法学 FlexASP 项目团队,主要研究传输触发架构(transport triggered architectures, TTA)及其协同设计工具链(TTA-based co-design environment, TCE)。

(3) 加州大学埃尔文分校(University of California at Irvine, UCI)嵌入式计算机系统中心(the Center for Embedded Computer Systems, CECS),主要从事嵌入式计算机体系结构及其设计方法学研究。

(4) 国内的著名研究组织有:中国科学技术大学研究团队从事 ASIP 设计方法学、模型验证及设计框架等方面的研究;国防科技大学则在体系结构,特别是 TTA 的 ASIP 及 SoC 系统实现方面做出较大贡献。

相关研究概述如下。

1. ASIP 设计空间探索

中国科学技术大学研究团队在对具体的 ASIP 设计步骤和现有设计方法学进行分析的基础上,提出一种“描述-搜索-细化”的 ASIP 体系结构设计流程,研究了体系结构描述、搜索与低功耗优化的问题,并依此实现了完整的设计环境原型^[5]。构建的 ASIP 设计平台 A²IDE 由以下几个主要部分组成:应用分析器 xpISR、GUI 工具界面 xpTOOLS、体系结构描述语言 xpADL、综合工具 xpSYN、生成工具 xp-GEN、可重定向编译器 xpCOM、仿真评估器 xpSIM 等。提出的 xpMODEL 模型以指令行为作为设计核心,引入指令行为有限状态机建立指令操作的时序调度机制,通过原语操作完成行为中对处理器资源的调用,有效地对指令的流水化操作进行建模。

亚琛工业大学 ICE 研究团队提出规约驱动的体系结构可配置 ASIP,即 rASIP 设计框架^[6]。将整个设计空间探索框架分为两个阶段:前制品(pre-fabrication)阶段和后制品(post-fabrication)阶段。前者通过软件工具生成、自由代码空间探索和 RTL 综合功能选择指令、接口、本地译码和体系结构;后者主要决定定制指令综合及减少应用运行时间。rASIP 很好地找到了性能和灵活间的平衡,实现了可配置处理器的自动指令及扩展综合,满足面向用户和处理器的约束,其中 ISE(指令集扩展)包括若干定制指令集。主要设计步骤有应用代码分析、CI(定制指令)定义与实现、软件适配器和工具生成、硬件体系结构实现。结合业界设计工具 LISATek 和学界开放平台 MPARM,力图使 SoC 设计者全方位地扫描体系结构配置空间,特别是多处理器 ASIP 设计空间探索。

2. TTA 的 ASIP 设计

采用 TTA 模板,可以对 ASIP 的重要方面,即专用指令自动扩展进行研究。国防科技大学研究团队提出了一种基于 TTA^[7]的嵌入式 ASIP 设计方法。根据用户提供的目标程序或者目标程序集合,进行嵌入式微处理器的设计和优化,并且可以按照不同的体系结构进行不同的并行代码的生成,在保证软件工具链兼容的同时,克服了 ASIP 设计过程中软件可重定向编译的困难,并进一步给出了 TTA 指导下的专用指令集处理器自动生成方法,有效地解决了指令集生成、可重定向编译和微结构设计等问题。

坦佩雷理工大学研究团队提出的工具链 TCE^[8]基于 TTA 的定制处理器模板,支持从应用源代码到综合处理器描述及机器代码优化的自动设计流程,可以有效地生成程序映像和处理器 HDL 综合。TCE 扩展版 TCEMC 支持同构多核仿真及其体系结构描述扩展,增加了任务及并行探索。仿真表明多核 ASIP 模板和线程运行库具有很大潜力。多核 ASIP 设计方法学探索将并行编程语言作为应用输