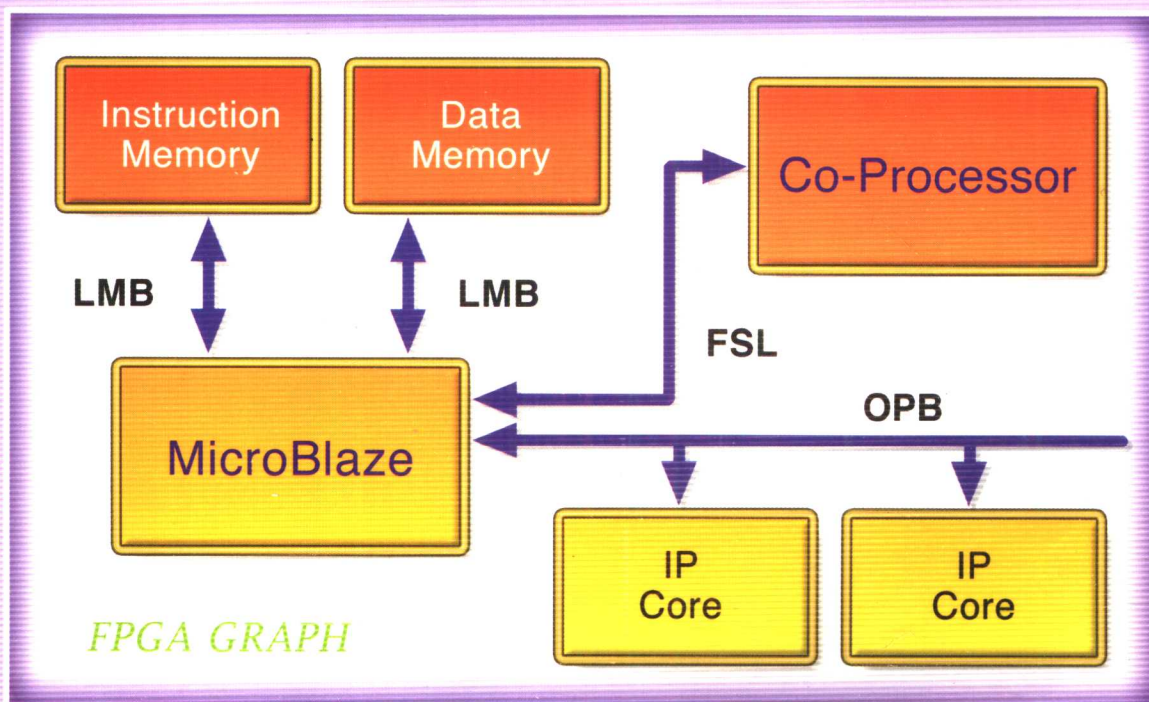


FPGA

嵌入式系统设计

孟宪元 钱伟康 编著



内 容 简 介

本书基于全球最大的可编程逻辑器件生产厂商美国 Xilinx 公司的系列 FPGA 产品,系统介绍了最新 FPGA 的内部结构、设计流程和 ISE8.1i 开发工具;全面介绍了 FPGA 实现嵌入式系统的基本原理和利用 FPGA 实现嵌入式系统的设计方法。在详细介绍硬件描述语言 VHDL 的基础上,如何利用 VHDL 语言设计数字系统,通过综合在 FPGA 上实现所设计系统的功能;利用可执行技术条件基于模型的设计方法,对设计的 DSP 系统进行建模、仿真和优化,以及通过代码生成,自动转换到硬件实现的过程;利用嵌入式处理器软核构造和裁剪嵌入式系统的设计方法,在单个 FPGA 芯片上利用软、硬件协同设计的方法,合理地配置资源,实现多处理器的嵌入式系统。

本书立足于工程实践,结合作者多年嵌入式系统和可编程逻辑器件的开发和教学经验,内容丰富,取材新颖,可为相关专业工程技术人员、研究生在设计和使用嵌入式系统时提供参考,也可作为高等学校电子电气信息类、仪器仪表类、自动化类及其他相关专业高年级本科生的教学参考书。

本书相关的 VHDL 设计例程和 DSP、嵌入式处理器的设计实例可到华信教育网 (<http://hxedu.com.cn>) 上下载。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

FPGA 嵌入式系统设计/孟宪元,钱伟康编著. —北京:电子工业出版社,2007.10

ISBN 978-7-121-05031-2

I. F… II. ①孟… ②钱… III. ①微型计算机—系统设计 ②可编程序逻辑器件
IV. TP360.2 TP332.1

中国版本图书馆 CIP 数据核字 (2007) 第 142195 号

责任编辑:张 榕 特约编辑:李云霞

印 刷: 北京市李史山胶印厂
装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 30 字数: 768 千字

印 次: 2007 年 10 月第 1 次印刷

印 数: 5000 册 定价: 49.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

半导体技术一直遵循著名的摩尔定律持续地发展,回顾半导体的发展历史,当一种技术具有可编程特性时,它就会处于支配的地位。目前,FPGA 器件不仅是逻辑门、布线资源和 I/O 可编程,随着半导体工艺采用 90 nm 和 65 nm,乃至不久将推出的 45 nm 工艺,在 FPGA 中增加了许多硬件核,如存储器、时钟管理和算法功能,直到增加嵌入式处理器硬核和软核等,也就是在 DSP 和嵌入式处理器等关键领域发展可编程技术,FPGA 已经成为在单片器件上同时提供可编程逻辑、高性能 DSP 和嵌入式处理器的系统级 (SoC) 芯片。在密度和性能提高的同时,成本和功耗也极大地降低,使得 FPGA 在更多的应用领域成为市场的主流。

FPGA 技术的关键在于用户可编程特性,利用 FPGA 作为高度并行处理引擎来实现 DSP 系统可以提供极高性能的信号处理能力,以及利用 FPGA 实现嵌入式 SoC 系统可以提供的一系列嵌入式处理器,使用相同的外设和 IP,在单个芯片上可以无缝地协同工作,同时与逻辑功能和 DSP 算法很好地结合。而且利用 FPGA 设计嵌入式系统可以满足样机开发周期短和产品上市快的要求,方便和快速地构造和仿真很复杂的系统,随着用户需要的变化或设计错误的发现,可以随时升级或修改。目前,围绕嵌入式系统展开研究和开发,已经成为计算机软硬件技术发展最有潜力的方向之一。利用 FPGA 设计嵌入式系统已经成为制造业的核心技术,并被广泛地应用到工业控制、系统仿真、医疗仪器、信息家电、通信设备等领域。

由于利用片上系统 SoC 集成的终端产品的复杂度不断增长,以及传统的软、硬件部件和设计方法的限制,使得设计复杂 SoC 器件要求以处理器为中心的设计新方法,基本的积木块是可配置的微处理器核,FPGA 能够提供一系列嵌入式处理器硬核和软核,可以满足设计复杂 SoC 器件的需要。在设计大型实时嵌入式信号处理系统时,将要求的技术条件转换成可执行的模块,这些模块转换成计算和通信的应用程序接口,由于 FPGA 是硬件实现 DSP,能够满足高性能的信号处理系统提出的技术要求,可以利用 MATLAB 的基于模型的设计方法由可执行模块进行系统建模、仿真和优化,并生成硬件实现的描述或网表进行综合和实现。本书重点讨论了这些基本的设计原理及实现方法,为更复杂系统的设计提供了较有价值的参考。

本书在介绍基于 Xilinx-FPGA 为设计平台的实现嵌入式系统的基本原理的基础上,全面讲述了利用 FPGA 实现嵌入式系统的设计方法。在介绍硬件描述语言 VHDL 的基础上,详细阐述了如何利用 VHDL 语言设计数字系统,并通过综合在 FPGA 上实现所设计系统的功能;引入了利用可执行技术条件的基于模型的设计方法,对设计的 DSP 系统进行建模、仿真和优化及通过代码生成,自动转换到硬件实现的过程;同时还深入介绍了利用嵌入处理器软核构造和裁剪嵌入式系统的设计方法,在单个 FPGA 芯片上利用软、硬件协同设计的方法,合理地配置资源,实现多处理器的嵌入式系统。

本书的作者较早接触 FPGA 技术,并深切感受到 FPGA 技术是数字系统设计的最佳途径。但是,目前专门讲述 FPGA 设计及应用方面的中文书籍,尤其是关于 Xilinx-FPGA 的最

新资料还很少,更多情况下需要设计人员直接查找和阅读相关英文资料。然而FPGA设计技术是一项实践性非常强的专业技术,需要一定的经验积累。为了能给更多的相关设计人员提供有益的参考并尽快掌握FPGA设计技术,作者查阅和参考了大量的中、英文资料,并根据为中国电子学会教育部、国防科学技术工业委员会信息中心等在各地举办的“FPGA嵌入式系统”培训班所涉及的研讨内容,以及参加由Xilinx公司组织的各种培训班学习的基础上,结合科研和工程实践经验,总结在学校教学和研究生课程等多方面积累整理后写成此书。本书在内容上尽量涉及相应技术的最新发展和最新的软件和硬件。

由于利用FPGA实现嵌入式系统是一项非常实用的技术,本书在介绍基本原理的基础上,对使用的设计软件进行了详尽的介绍,便于读者进行实践和操作。除了本书的内容之外,在电子工业出版社的华信教育网上(<http://hxedu.com.cn>),以及与非门电子产业社区(<http://www.eefocus.com>)还提供了数字系统设计、DSP系统设计和嵌入式处理器设计等有关的设计实例,供读者参考。考虑读者可能有不同的实验硬件,网站上也提供一些实验硬件的参考材料和设计例子,包括清华大学科教仪器厂生产的低价位的THX3E-1实验板。在Xilinx公司资助的中国首个FPGA开源硬件社区——Openlhard(www.openhard.org),可以交流开发FPGA嵌入式系统的经验和体会。

由于电子技术发展极其迅速,本书介绍的内容是有关最新技术的基本概念和主要的设计方法及使用操作,更深入的技术内容和开发方法限于篇幅没有列入本书的范围,有待读者熟练后再深入和掌握。

本书在编写过程中,得到Xilinx公司上海代表处的吴晓东、梁龙和谢凯年等先生的大力支持,提供丰富的资料,并提出了宝贵的建议,在此表示衷心的感谢。

清华大学电子工程系和上海理工大学电气工程学院的有关教师、研究生和本科生为本书的出版给予了大力的支持和关照,他们是张卫杰、贾志鹏、邹祥、孙智、周伟、施顺国、刘涛、李丹、谢明、杨滨、刘义红、郭强、李秋实、刘洋等,对于他们为本书编写、校对、硬件调试和软件编程等方面所做的工作表示感谢。

由于作者的水平和能力有限,编写时间仓促,对于有关新技术的掌握和理解有局限,嵌入式系统的相关技术发展迅速,软件和器件的更新快,应用领域广,书中的不足和错误之处恳请读者批评指正。

孟宪元 mengxy@mail.tsinghua.edu.cn

钱伟康 elelsec@usst.edu.cn

目 录

第 1 章	FPGA 实现嵌入式系统概述	1
1.1	嵌入式系统概述	1
1.1.1	嵌入式系统的组成	1
1.1.2	嵌入式片上系统	3
1.2	FPGA 嵌入式系统的特性	6
1.2.1	FPGA 的基本特性	6
1.2.2	FPGA 实现嵌入式系统的优势	8
1.2.3	FPGA DSP 嵌入式系统	9
1.2.4	FPGA 微处理器嵌入式系统	12
第 2 章	嵌入式系统设计	17
2.1	数字系统的抽象表示	17
2.2	系统设计的方法	21
2.3	基于模型的系统设计方法	24
2.4	IP 核	27
2.5	电子系统级——ESL 设计	32
2.6	AccelDSP 软件工具	35
第 3 章	VHDL 及编程技巧	37
3.1	VHDL 简介	37
3.1.1	关于 VHDL	37
3.1.2	VHDL 代码的可综合性	38
3.1.3	用 VHDL 语言设计的流程	40
3.1.4	设计样例	41
3.2	VHDL 代码基本结构	41
3.2.1	用户定义及库声明	42
3.2.2	实体 (ENTITY) 描述	44
3.2.3	结构体 (构造体 ARCHITECTURE) 描述	45
3.2.4	基本对象、数据类型及运算符	47
3.2.5	并行语句结构	52
3.2.6	顺序语句结构	56
3.2.7	子程序及其引用	63
3.2.8	包集合与库	65
3.2.9	元器件配置	66
3.2.10	电路设计实例	69
3.3	VHDL 编码风格	82
3.3.1	用 VHDL 语言设计时提高硬件综合效率的主要策略	83
3.3.2	优化系统速度的 VHDL 语言设计策略	84
3.3.3	面积优化的 VHDL 语言设计策略	87

3.3.4 提高设计性能的编程技巧	90
3.4 VHDL 设计例程参考	94
第 4 章 FPGA 的功能块硬核	95
4.1 系统设计的同步时钟	96
4.1.1 同步设计基本原理	96
4.1.2 SoC 中的 PLL	100
4.2 FPGA 时钟电路	105
4.2.1 Xesium 全局时钟	105
4.2.2 数字时钟管理 (DCM)	108
4.2.3 相位匹配时钟分频器 (PMCD)	114
4.2.4 ChipSync 源同步	116
4.2.5 RocketIO 高速串行接口	121
4.3 系统设计的逻辑资源	125
4.3.1 逻辑资源的组成结构	125
4.3.2 查找表 (LUT) 的逻辑功能	128
4.3.3 存储元件	130
4.4 系统芯片的存储器	130
4.4.1 片内 RAM	130
4.4.2 分布 RAM	132
4.4.3 块 RAM	132
4.4.4 片外存储器接口	144
4.5 输入/输出模块	145
4.5.1 输入/输出接口标准	145
4.5.2 数字控制阻抗 (Digitally Control Impedance, DCI)	149
4.5.3 输入/输出数据寄存器	153
4.5.4 数据的 SERDES	159
第 5 章 数字系统设计流程 (Xilinx ISE8.1i 简明教程)	165
5.1 ISE Foundation 软件简介	165
5.1.1 Xilinx ISE8.1i 软件特性	166
5.1.2 ISE8.1i 软件的安装和使用环境	167
5.1.3 FPGA 设计流程	175
5.2 用 VHDL 语言设计输入	177
5.2.1 创建一个新工程	178
5.2.2 创建一个 VHDL 代码的计数器源文件	181
5.2.3 利用计数器模板向导生成 VHDL 源代码设计	182
5.3 设计综合	183
5.3.1 查看综合报告	184
5.3.2 查看综合后的 RTL 视图	185
5.3.3 综合工具高级选项	186
5.4 设计仿真	188
5.4.1 功能仿真 (Behavioral Simulation)	188
5.4.2 时序仿真 (Post Route Simulation, 布局、布线后仿真)	192



5.5 设计实现	193
5.5.1 转换 (Translate)、映射 (Map) 和布局、布线 (Place & Route)	193
5.5.2 设计实现的运行步骤	194
5.6 CPLD/FPGA 编程	195
5.6.1 JTAG 编程	196
5.6.2 Xilinx-FPGA 串、并行主/从模式编程	201
5.7 ISE8.1i 高级设计技巧	212
5.7.1 布局、布线器 (Floorplanner)	212
5.7.2 约束编辑器 (USER Constraints Editor)	215
5.7.3 FPGA 编辑器 (FPGA Editor)	225
5.7.4 片内逻辑分析仪工具——ChipScope™ Pro 8.1i 的使用	231
第 6 章 FPGA DSP 嵌入式系统	241
6.1 DSP 的一般特性	241
6.2 FPGA 实现 DSP	243
6.3 FPGA 的 DSP 硬件资源	246
6.3.1 逻辑资源的 DSP 特性	246
6.3.2 乘法器专用模块	248
6.3.3 DSP 专用模块	249
6.4 FPGA 实现 DSP 的结构变换	253
6.4.1 迭代限界 (Iteration bound)	253
6.4.2 重新定时 (Retiming)	255
6.4.3 流水线	258
6.4.4 并行处理	261
6.5 FPGA 实现 DSP 的算法变换	262
6.5.1 “分布算法”	262
6.5.2 CORDIC 算法变换	265
6.5.3 CIC (Cascaded Integrator Comb) 滤波器	269
6.6 FPGA 的 DSP 应用	271
6.6.1 MAC 引擎的 FIR 滤波器	271
6.6.2 完全并行的滤波器结构	274
6.6.3 转置 FIR 滤波器	275
6.6.4 抽取滤波器和插值滤波器	277
6.6.5 多相滤波器	278
第 7 章 FPGA DSP 嵌入式系统设计	284
7.1 设计流程概述	284
7.2 FPGA 设计 DSP 技术	288
7.2.1 浮点数与定点数的表示与转换	288
7.2.2 采样周期的设置	291
7.3 System Generator 的基本模块	292
7.3.1 System Generator 模块	292
7.3.2 Black Box 模块	293



7.3.3	ModelSim 模块	295
7.3.4	Gateway In 模块和 Gateway Out 模块	296
7.3.5	Concat 模块、Convert 模块、Reinterpret 模块和 Slice 模块	296
7.3.6	模块通用属性	298
7.4	设计 FIR 滤波器	299
7.4.1	产生 FIR 滤波器的系数	299
7.4.2	输入 FIR 滤波器系数	301
7.4.3	在 Simulink 中仿真 FIR 滤波器	302
7.4.4	完成 FIR 滤波器设计	304
7.4.5	实现 FIR 滤波器	304
7.4.6	利用资源估计器估计设计	305
7.4.7	执行硬件在环路中的校验	305
7.4.8	连接演示板, 通过 Simulink 仿真 FIR 设计	306
7.5	利用协同仿真校验 MAC FIR 滤波器	307
7.5.1	产生 MAC FIR 核	307
7.5.2	编写 Black Box 模块封装文件	310
7.5.3	为 HDL 协同仿真完成 MAC FIR 的设计	312
7.5.4	连接 Black Box 图标	314
7.5.5	执行 HDL 协同仿真	315
7.5.6	执行硬件在环路校验	316
7.5.7	连接演示板, 通过 Simulink 仿真 MAC FIR 设计	317
7.6	设计 MAC FIR 滤波器	318
7.6.1	分析系数	319
7.6.2	添加控制逻辑并对它参数化	320
7.6.3	添加双口 RAM	322
7.6.4	在数据端口添加填充位和去填充位	324
7.6.5	完成 MAC FIR 设计	326
7.6.6	用各种信源测试设计	328
7.6.7	实现 MAC FIR 设计	330
7.6.8	执行硬件在环路校验	331
第 8 章	FPGA 嵌入式处理器	337
8.1	MicroBlaze 微处理器软核	337
8.1.1	MicroBlaze 嵌入软核	337
8.1.2	流水线结构	341
8.1.3	存储器结构	342
8.1.4	指令 Cache 和数据 Cache	342
8.2	嵌入式系统开发包——EDK	344
8.2.1	EDK (Embedded Development Kit)	344
8.2.2	XPS (Xilinx Platform Studio)	345
8.2.3	基本系统构造 (BSB) 向导	347
8.3	微处理器系统硬件设计	349
8.3.1	硬件设计环境	349



8.3.2 存储器组合	350
8.3.3 总线结构	350
8.3.4 IP 核	351
8.3.5 硬件设计结果	354
8.3.6 硬件设计实现	356
8.4 嵌入式处理器软件设计	357
8.4.1 GNU 工具	358
8.4.2 器件驱动	359
8.4.3 中断	361
8.4.4 库	361
8.5 MicroBlaze 诊断环境	362
8.5.1 GNU Debugger-GDB 工具	362
8.5.2 Xilinx Microprocessor Debugger, XMD	364
8.6 XPS 软件开发包 (SDK)	366
第 9 章 FPGA 嵌入式微处理器设计	371
9.1 设计流程概述	371
9.2 EDK 软件安装	372
9.3 FPGA 嵌入式处理器硬件设计	373
9.3.1 硬件设计——利用基本系统构造框架	373
9.3.2 硬件设计——利用 Add/Edit Cores	383
9.3.3 硬件设计——添加定制的 IP	389
9.4 FPGA 嵌入式处理器软件设计	401
9.4.1 软件设计——编写基本应用软件	401
9.4.2 软件设计——编写高级应用软件	412
9.4.3 软件开发包 SDK	422
9.5 MicroBlaze FIR 协处理器设计	431
9.5.1 产生 FSL IP 核	432
9.5.2 开发 FSL IP 核	437
9.5.3 输入 FSL IP 核	441
9.5.4 MicroBlaze FIR 协处理器	447
附录 A sysytem.mhs 文件	460
参考文献	466

1.1 嵌入式系统概述

1.1.1 嵌入式系统的组成

在许多领域中广泛应用的嵌入式计算系统简称为嵌入式系统,它是在更大的电子器件或设备中嵌入的重复完成特定功能的计算系统,经常不被器件的使用者所识别,但在各种常用的电子器件中能够找到。例如,消费类电子产品中的手机、寻呼机、数码相机、摄像机、录像机、个人数字助理等,家用电器中的微波炉、洗衣机、烤箱、门禁系统、照明系统等,商用装置中的卡片读入器、扫描仪、打印机等。这些嵌入式系统具有以下几个共同的特性。

(1) 单一的功能。一个嵌入式系统通常重复地执行一个特定的程序,除非嵌入式系统的程序被新的版本更新,或者由于尺寸的限制几个程序在一个系统中被调进或调出,即使这些例外也表示系统具有特定的功能。

(2) 严紧的限制。所有的计算系统在设计时都有许多约束的要求,但是对嵌入式系统的约束特别严紧,如成本、尺寸、性能和功耗等设计的限制条件可以评定系统实现的特性,而嵌入式系统有特别高的设计技术要求。

(3) 实时的反应。许多嵌入式系统必须连续地对系统环境的变化做出反应,并且无延误地实时计算出确定的结果。

(4) 软、硬件共存。一般嵌入式系统由嵌入式处理器、系统的外围硬件设备、嵌入式操作系统和系统应用软件四部分组成,是一个以微处理器为核心的数字系统,涉及硬件和软件两方面的技术。嵌入式系统的微处理器可以是通用处理器或微处理器、单片机或 DSP 的专用处理器、单用途处理器等。

由于设计的度量标准是针对嵌入式系统实现的特性和功能进行量测,嵌入式系统的设计架构必须实现全部预定的功能,但是面临的设计挑战是同时要对大量的设计度量标准进行优化。

除了不可回收的工程(NRE)成本外,对设计系统的要求包括单元成本、尺寸、性能、功耗、灵活性、样机时间、上市时间、可维护性、故障率和安全性等。

因此,嵌入式系统可以定义为“以面向应用为中心,软件、硬件可裁剪的完成特定功能的可计算设备,适应应用系统对功能、可靠性、成本、体积、功耗等综合性严格要求的专用计算机系统”。



大部分嵌入式系统不使用实时操作系统(RTOS)或任何商业系统软件,这曾经被认为是相当合理的事情。当软件在整个工程量中只占到小份额时,比起寻求最佳的软件开发和配置环境来,还有更加值得花费时间和金钱的地方。但是,情况已经发生了变化。现在,软件已经成为嵌入式系统整个工程的主要开销。这使得有可能通过软件去提高工程的效率和加快进入市场的进度。设计效率和进度的改善来自优良的方法学、好的工具,以及有才能的编程人员。

操作系统是一种开发工具,也是最终产品的一部分。选择系统软件关系到开发成本、进入市场的时间及单位成本,这是值得加以关注的。

RTOS 具有两种确定的属性:①它是一个操作系统;②它有某种程度的实时性能。这两种属性并存还比较罕见。尽管有几十种操作系统存在,但它们中只有小部分具有实时性能。实时内核有百种以上,但 RTOS 只有少数几种。

从最一般的意义上说,RTOS 是隐藏计算机硬件特征复杂性的软件。这囊括了广泛类型的系统软件。几乎所有的 RTOS 都提供线程,造成处理器在同一时间能做几种工作的假象。而且,大多数系统软件都提供把 I/O 包装成如 `read()`、`write()`、`open()` 和 `close()` 一类函数的服务。

只提供比线程和基本 I/O 略多一点的系统软件一般称做小内核。现在,领先的商业内核有 Nucleus 和 Wind 等。

操作系统超出内核的地方在于它提供保护。操作系统在进程中运行应用软件。它预见进程中的错误,使进程除了自身外不会影响到其他方面,而且当进程意外地中止后,它能顺利地恢复。现在,领先的实时操作系统有 OS-9、QNX 和 LynxOS 等。

实时软件是为需要满足时限的系统而设计的。一切精心编写的系统软件都关心性能,而实时系统软件还有更为复杂的目标:

- (1) 它应该能够精确地预测系统执行时间。容许的误差限与硬件有关,可能是 $200\ \mu\text{s}$ 或 $2\ \mu\text{s}$ 。非实时操作系统与预期的时限可能相差 $1\ \text{ms}$ 甚至更多。
- (2) 它应该反应灵敏。RTOS 工程师尽最大努力去缩减环境切换与中断响应时间。
- (3) 它应该快速运行。

这 3 个性能要求是按照重要性大小顺序排列的。传统的操作系统把重要性顺序恰好调个头,几乎是只关心典型的性能而不顾其他。

对嵌入式控制而言,实时操作系统常常是唯一的实际选择,特别当多个控制环路的运行状况在可控优先级下必须能够被预测时,更是如此。

嵌入式软件正在成为世界软件行业抢占的新战略高地。嵌入式软件的应用领域非常广,市场分得非常细,产品更注重个性化需求,每个领域都可以拥有各种各样的专用嵌入式软件。不同领域、不同产品,需要不同的微型计算机芯片,需要装入各种不同的应用软件,也需要不同的嵌入式操作系统做支撑。

嵌入式操作系统领域尚未形成像微软公司对 PC 操作系统那样的垄断,发展中国自己的嵌入式操作系统,将带动一批 SDK、数据库、支持工具和应用软件,推动国内软件产业的整体发展,中国软件业将一改在台式机软件市场的被动局面。

嵌入式系统的硬件性能同样也在不断提高,尤其是 $90\ \text{nm}$ 和 $65\ \text{nm}$ 半导体工艺技术的芯片上市,使得硬件平台的处理能力极大地提升,器件价格直线下降,嵌入式系统已经成为产品数字化改造、智能化增值的关键性技术措施,也加快了信息技术在自主创新与产业化程度方面



的进程。

进入 21 世纪后,在新一轮通信技术、工业技术、汽车电子、消费电子、生物和医疗仪器、军事和航天设备等行业巨大智能化、数字化、集成化装备市场的需求下,全球的嵌入式软件及系统产业得到了飞速发展,可以说已经广泛地应用到人类生活的方方面面。据统计,2004 年全球嵌入式系统产品的产值已达到 2000 亿美元,2005 年全球 MCU 的市场达到 125 亿美元,而 DSP 市场达到 84 亿美元,估计全世界嵌入式系统产品的潜在市场将超过 10000 亿美元。

由于我国具有雄厚的制造业基础,与传统的制造业结合,嵌入式系统有着巨大的市场空间。同时,基于嵌入式系统的应用开发和技术创新,将推动新产品、新产业的诞生。

2004 年,我国嵌入式系统应用产品的经济总量超过 1000 亿元,其中嵌入式处理器芯片约为 120 亿元。2004 年,我国嵌入式微处理器销售总量大约 13 亿片。嵌入式 DSP 市场已无处不在,市场规模已接近通用 DSP 的两倍。增长的势头强劲,其应用包括 DVD 播放机、机顶盒、音/视频接收设备、MP4 播放机、数码相机和汽车电子等各个领域。

因此,发展嵌入式系统产业,包括嵌入式操作系统的嵌入式软件和各个应用领域的嵌入式硬件设备,将是我国提出电子信息产业实现由“中国制造”向“中国创造”迈进的重要契机和突破口,成为我国信息产业增长由粗放型向节约型转变,实现可持续发展的重要途径。

1.1.2 嵌入式片上系统

半导体集成电路是通过微细加工技术直接制作在半导体单晶圆片上的电子电路。从 1958 年世界上第一块集成电路诞生至今的 49 年中,世界集成电路技术与产业飞速发展,历经小规模(数百元件)、中规模(数千元件)、大规模(数十万元件)、超大规模(百万元件)的发展阶段,到今天已进入特大规模(千万以上元件)时代。

随着集成度的提高和电路规模的增大,电路中单元器件尺寸不断缩小,图形特征尺寸成为每代集成电路技术的特有表征。这一发展过程一直遵循著名的“摩尔定律”和“按比例缩小定理”:每 2~3 年集成度提高两倍,器件特征尺寸按比例缩小,工艺技术相继更新。目前,国际主流生产工艺技术为 130 nm, 90 nm 和 65 nm 技术也将进入大生产阶段。

半导体集成电路产业是一种典型的技术驱动型产业,其快速更新换代的技术动力来源于工艺技术的快速发展。由于器件尺寸不断按比例缩小,使得集成电路规模和性能在成倍提高的同时能够保持成本的稳定,从而使得集成电路产品的更新能够迅速地为社会所接受。这直接导致了全球半导体市场规模的急速扩张,引发了当代以电子信息技术为核心的新一轮科技革命,集成电路也因此成为当今“全球化”和“知识经济”中最重要的物质基础。而半导体工艺技术则成为目前人类加工技术中精度与难度最高、技术最为密集,同时也是进步最快的一种系统性的工程技术。

集成电路芯片的制造有数百道工序,由数十种上千套设备仪器完成,而微细加工技术是半导体工艺的核心技术,始终是半导体技术发展水平的先导和标志。在几十年的发展历史中,微细加工技术不断地突破人们根据当时现状为之设想的加工“极限”,总是根据摩尔定律提出的时间节点获得满足需求的解决方案,从而造就了所谓的“摩尔定律神话”。

在 20 世纪 90 年代中期,人们曾经为 100 nm 和 50 nm 是否会成为集成电路技术的又一个“极限”而展开过一场全球性的热烈讨论,而今天,半导体工艺技术进入 90 nm,这标志着人类的微细加工能力即将进入又一个空前的高度,整个半导体领域的前沿热点从制造技术、器件物



理、工艺物理到材料技术等各方面随之全面进入纳米领域。可以预见,在今后相当长的一段时间内,集成电路的特征尺寸还会进一步按比例缩小,集成度还将进一步增加,尽管需要不断克服从设计、制造到封装、测试的一系列前所未有的“极限”挑战。但在全球巨大规模的密集资金和人力投入的保证下,半导体工艺技术仍将按照“摩尔定律”的步伐继续发展。

预计到 2010 年,半导体工艺技术将进入 50 nm 以下,而最终可能达到 10 nm,从而使电路集成度达到 1000G(千吉)级晶体管数。

今天,人们更关注的是半导体工艺技术进入 90 nm 以下后,如何充分利用这一制造能力,设计出功能更强大的集成电路产品,以迎接全球信息技术的又一轮发展高潮。在经历了几十年持续快速发展之后,微电子产业将逐步从新兴产业发展成为成熟产业,其市场发展的主流将逐步从追求核心产品的技术提升,转向各个领域的全方位应用,以满足各个领域的集成化、信息化和智能化的需求。而半导体工艺技术的进步将为继续这一进程提供制造技术保障。

但半导体工艺技术在进入 90 nm 以下后,最终将会带来怎样的影响,现在还很难做出全面的估计。作为对这一技术能力的直观描述,可以设想,随着制造工艺技术和系统芯片(System-on-Chip)设计技术的发展,将可以制造出一种集成了无线通信、个人计算机、多媒体功能的单芯片个人信息终端,从而在技术上保证个人在任何时间、任何地点都可以成为全球信息网络的终端。

随着半导体工艺技术的发展,IC 设计者能够将越来越复杂的功能集成到单硅片上。SoC(片上系统)正是在集成电路(IC)向集成系统(IS)转变的大方向下产生的。从狭义角度讲,SoC 是信息系统的芯片集成,是将系统集成在一块芯片上;从广义角度讲,SoC 就是一个微小系统。如果说中央处理器(CPU)是大脑,那么 SoC 就是包括大脑、心脏、眼睛和手的系统。SoC 的出现使集成电路发展成为集成系统,整个电子整机的功能将可以集成到一块芯片中。在不久的将来,集成电路与电子整机之间的界限将被彻底打破。

SoC 就是将微处理器、模拟 IP 核、数字 IP 核和存储器(或片外存储控制接口)集成在单一芯片上。它通常是客户定制的(CSIC),或是面向特定用户的标准产品(ASSP)。

SoC 是面向特定用户的,能最大满足嵌入式系统要求的芯片,因而具有很多优势:能极大地改善功耗开销;可减少印制板上的部件数和引脚数;可减少板卡失效的可能性;有利于板卡的性能改善(由于片内连线缩短);可降低风冷要求;可减少系统开发成本;尤其适合数字化产品开发,如手持设备、信息家电等。

从分立元件到集成电路再到片上系统,是微电子领域的几次革命。21 世纪,集成电路将进入 SoC 时代。

1. SoC 发展中的焦点技术

SoC 设计准入的最大门槛是专门技术、IP 库和 SoC 总线架构支持,需要广泛的多功能 IP 和将客户逻辑与之集成在一起的设计艺术,以满足客户产品开发的需求。由此许多第三方 IP 供应商得到快速发展,他们的成功要么具有独一无二的且极具价值的 IP,要么具有良好声誉的库。SoC 设计者通过“重用”证明的 IP,不仅利用最新工艺技术优势,而且减少了开发周期和风险。

SoC 的发展离不开应用领域的需求牵引。在进行片上系统设计时,首先要考虑的问题是系统的体系结构。为了提高开发模块的重复利用率、降低开发成本,用户采用了 SoC(芯片内

部)总线、芯片间总线(如 SPI、I²C、UART、并行总线)、板卡间总线(如 ISA、PCI、VME)、设备间总线(如 USB、1394、RS-232)。SoC 总线为用户提供了一个堪称“理想”的环境:片上系统模块之间不会面临干扰、匹配等传统问题,但是片上系统的时序要求却异常严格。

由于 OpenCore 和其他致力于开放知识产权(Open Intellectual Property)组织的大力推广(开发设计了大量基于标准化片上总线的免费模块),用户在片上系统总线的选择上更倾向于采用那些标准化、开放化的方案。目前,总线架构有很多种,包括 IBM 公司的 CoreConnect、ARM 公司的 AMBA、Silicore 公司的 Wishbone、MIPS 技术公司的 SoC-it 和 CoreFram 等。

SoC 的发展重点主要包括:

- (1) 总线结构及互连技术,直接影响芯片总体性能发挥;
- (2) 软、硬件的协同设计技术,主要解决硬件开发和软件开发同步进行问题;
- (3) IP 可重用技术,如何对其进行测试和验证;
- (4) 低功耗设计技术,主要研究多电压技术、功耗管理技术和软件低功耗利用技术等;
- (5) 可测性设计方法学,研究 eJTAG 设计技术、批量生产测试问题;
- (6) 超深亚微米实现技术,研究时序收敛、信号完整性、天线效应等。

SoC 将引领新一代嵌入式处理器的技术发展,它是以嵌入式系统为核心,集软、硬件于一体,并在系统集成中追求产品系统最大包容性,能成功实现多学科的合作与融合。SoC 设计技术为计算机专业人才介入 IC 设计领域提供了一个机会。不仅在 SoC 芯片设计上需要较强的计算机体系结构背景知识,而且 SoC 突出了软件开发的比重,需要计算机专业人士的介入,需要提供良好的开发平台和嵌入式操作系统。

SoC 的发展将不断满足日趋增长的功能密度、灵活的网络连接、轻便的移动应用、多媒体的信息处理等需求。SoC 需具备 LCD、USB、CAN、MAC/WLAN 或 IrDA 通信接口等,同时也需要提供相应的通信组网协议软件和物理层驱动软件,甚至浏览器。

SoC 将满足人们以 GUI 屏幕为中心的多媒体界面与信息终端交互需求,如手写文字输入、语音拨号上网、收/发电子邮件、传送彩色图形/图像及语言同声翻译等。SoC 将具有 32 位、64 位 RISC 芯片或信号处理器 DSP 等增强处理能力,同时支持嵌入式 RTOS 发展,采用实时多任务编程技术和交叉开发工具技术来控制功能复杂性,简化应用程序设计,保障软件质量,缩短开发周期。

2. SoC 是实现跨越式发展的桥梁

在过去 5 年中,SoC 得到了快速发展。据预测,SoC 销售额将从 2002 年的 136 亿美元,增长到 2006 年的 347 亿美元,年增长率超过 20%。

另外,世界芯片复杂度的年增长率为 58%,但设计能力的增长仅为 20%。由此看出,世界集成电路设计能力的增长远远跟不上芯片复杂度增长的速度,这为集成电路设计产业提供了难得的发展机会。面对集成电路向 SoC 的转型,我国实现集成电路设计业跨越的一个历史机遇已经来临。许多专家建议,我国应优先发展芯片设计业,特别重视 SoC 提供的发展机会。

相信在未来 3~5 年内,高端嵌入式处理器将以 SoC 的发展为代表,成为各相关学科的交汇点。在 SoC 相关学科领域中,应注意吸收与培养其他学科领域人才,如光、机、电等学科,不断改善 SoC 研究队伍组织结构,加强跨学科的 SoC 综合技术研讨,积极沟通观念、信息与技术,以培养 SoC 的跨学科高级人才。

只有通过跨学科的相互交融,才能促使 SoC 设计技术产生质的飞跃。SoC 必将导致又一次以片上系统为特色的信息技术革命,21 世纪初期将是 SoC 技术真正快速发展的时期。

1.2 FPGA 嵌入式系统的特性

1.2.1 FPGA 的基本特性

现场可编程门阵列(FPGA)已经从最初的可编程逻辑器件发展到当今的可编程系统,以其丰富的可编程资源成为系统设计的平台。早期的 FPGA 器件只提供三种基本的用户可编程逻辑资源,即

- (1) 可编程 I/O。器件的引脚可以由用户编程为输入或输出、TTL/CMOS 电平、是否接入上拉/下拉电阻等。
- (2) 可编程逻辑模块。包括实现组合逻辑和时序电路的基本逻辑单元,可以由用户编程来构造数字系统的各种电路。
- (3) 可编程互连线。由长线和分段的短线,以及开关电路组成的连线资源,可以由用户编程实现系统功能所要求的布线。

按照器件进行配置及实现组合逻辑的不同方法,可以将 FPGA 器件分为以下两大类型。

1. SRAM 查找表结构

SRAM 查找表结构的 FPGA 是通过对器件的 SRAM 配置存储器实现对器件的编程,由 SRAM 的特性决定了这类器件的功能是掉电丢失的,需要每次加电重新进行配置,因此可以通过加载电缆人工进行配置,或者从 PROM 或闪存等附加的存储器自动进行配置。SRAM 存储器无限制的配置次数是其最大的特点,也是得到广泛应用的主要原因。

查找表实现组合逻辑是将逻辑函数的真值表存入 SRAM 的存储器,输入信号作为查找表的地址,选择相应的函数值作为输出,因此存储器的存取时间决定了实现逻辑的速度,而与逻辑函数的复杂程度无关。图 1-1 说明了查找表如何实现一位全加器。

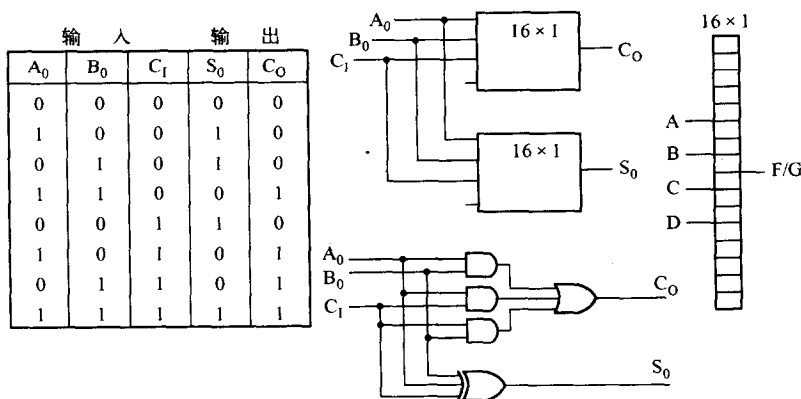


图 1-1 查找表实现一位全加器

由于查找表和配置存储器都是由 SRAM 存储器组成的,所以“SRAM 查找表结构”的 SRAM 具有双重的含义。图 1-2 所示为 FPGA 中的 SRAM 配置存储器。Xilinx、Altera 和 Lattice 三家公司的 FPGA 器件是 SRAM 查找表结构的。

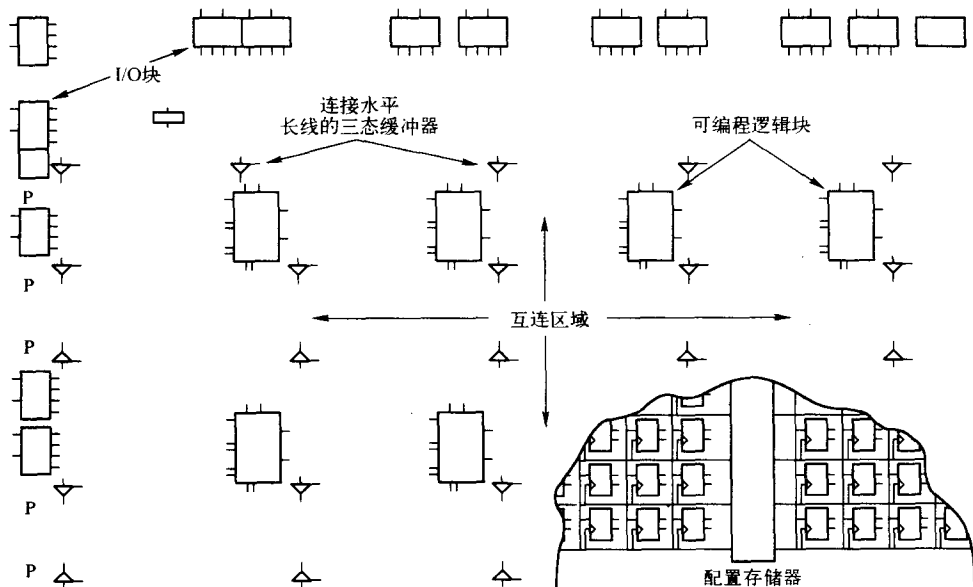


图 1-2 FPGA 中的配置存储器

2. 反熔丝多路开关结构

反熔丝多路开关结构的 FPGA 是通过反熔丝电阻实现对器件的编程。对器件编程前反熔丝电阻是高阻,相当于熔丝断开;而编程后反熔丝电阻是低阻,相当于熔丝接通。因为熔丝是编程前接通,编程后断开,所以反熔丝电阻的作用正好与 PROM 器件熔丝的作用相反,两者在编程之后都是不可再复原的,所以都是一次性的。Actel 和 QuickLogic 两家公司的 FPGA 器件都是采用反熔丝技术进行器件编程的。

多路开关实现组合逻辑是对多路开关的输入端和控制端接入信号,或电平后得到逻辑函数的输出。反熔丝多路开关结构的 FPGA 如图 1-3 所示。下面从 Actel 公司的 ACT-1 器件为例,说明如何实现一位全加器。

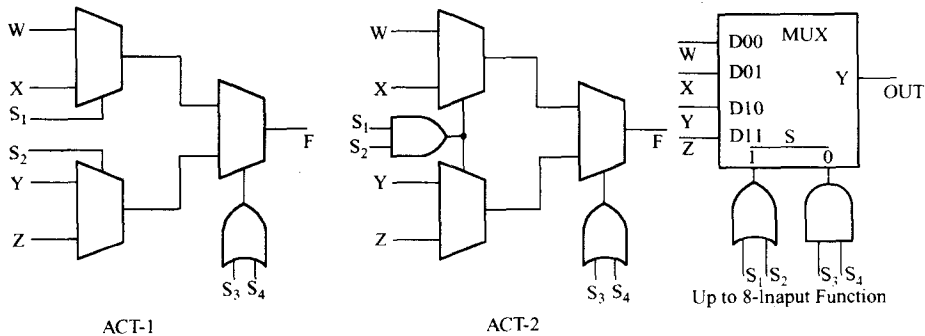


图 1-3 反熔丝多路开关结构 FPGA

ACT-1 由 3 个两输入多路开关和 1 个或门组成基本积木块,可以实现的逻辑函数为

$$F = (\overline{S_3 + S_4})(\overline{S_1}W + S_1X) + (S_3 + S_4)(\overline{S_2}Y + S_2Z)$$

设置每个变量为一个输入信号或一个固定电平时,可以实现 702 种逻辑函数。

例如,当设置为 $S_4=0, S_3=C_{in}, W=A, X=\overline{A}, S_1=B, Y=\overline{A}, Z=A, S_2=B$ 时,可实现全加器输出和信号 Sum 的逻辑函数:

$$\text{Sum} = \overline{C_{in}}(\overline{B}A + B\overline{A}) + C_{in}(\overline{B}A + BA) = \overline{C_{in}}(\overline{A} \oplus B) + C_{in}(\overline{A} \oplus B) = C_{in} \oplus A \oplus B$$

当设置为 $S_4=0, S_3=A, W=0, X=C_{in}, S_1=B, Y=C_{in}, Z=1, S_2=B$ 时,可实现全加器输出的进位信号 C_{out} 的逻辑函数:

$$C_{out} = \overline{A}(\overline{B_0} + BC_{in}) + A(\overline{B}C_{in} + B_1) = (\overline{A}B + A\overline{B})C_{in} + AB = AC_{in} + BC_{in} + AB$$

由于反熔丝多路开关结构 FPGA 是一次性编程的,所以本书主要针对 SRAM 查找表结构的 FPGA。利用 FPGA 是以集成电路的方式实现用户应用领域的专门设计,所以 FPGA 是可编程 ASIC。现在,电子系统设计者可以在实验室或办公室,甚至在应用现场设计出所需的专用集成电路,实现系统的集成,缩短样品开发、产品上市的时间,降低设计和开发成本。对于 SRAM 查找表结构的 FPGA 还具有静态可重复编程或在线动态重构的特性,使硬件的功能可以像软件一样通过编程来配置、仿真、修改,因此被称为“液体硬件”。

工艺进入深亚微米之后,FPGA 的集成度增加,可以拥有大量的逻辑资源。为了降低功耗,也为了满足系统设计的需求,更多的硬模块集成到 FPGA 中,包括块 RAM/ROM、嵌入乘法器、DSP 模块和带 PLL 或 DLL 的时钟电路;I/O 脚也分成几个组合,可以按照系统设计的要求设置成多种接口标准。当 FPGA 中集成了嵌入式处理器的软核或硬核时,已经具备了进行可编程系统设计平台的能力,并称为可编程片上系统——SoPC(System-on-Programmable Chip)。

1.2.2 FPGA 实现嵌入式系统的优势

从系统对上市时间的要求、可编程的特性及集成度等方面考虑,以可编程门阵列(FPGA)来实现可配置的嵌入式系统已越来越广泛。

图 1-4 所示为 FPGA 实现嵌入式系统的市场预测,估计从 2002 年的 23 亿美元,将增长到 2008 年的 63 亿美元。

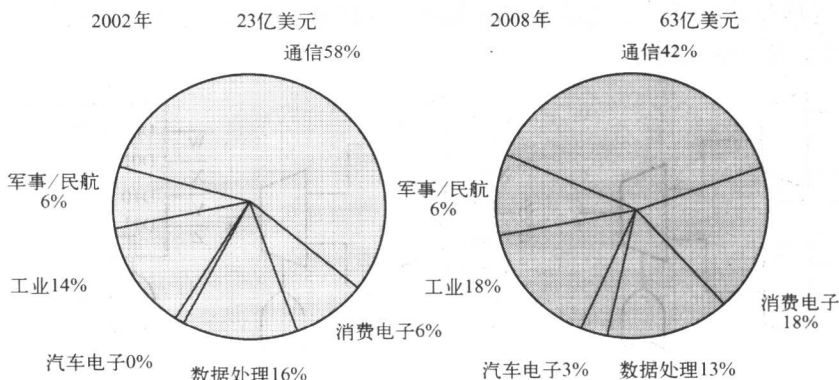


图 1-4 FPGA 实现嵌入式系统的市场预测