

EDA

EDA工具应用丛书

基于FPGA的 嵌入式开发与应用

徐光辉 程东旭 黄如 等编著
徐志军 审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



<http://www.phei.com.cn>

EDA 工具应用丛书

基于 FPGA 的嵌入式开发与应用

徐光辉 程东旭 黄如 等编著

徐志军 审

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书全面系统地介绍了基于 SOPC 的嵌入式系统设计技术, 内容包括 Altera 可编程逻辑器件硬件结构、Quartus II 开发软件的使用、SOPC 原理与设计实例, 以及基于 FPGA 的算法实现。

本书内容丰富, 取材新颖。可作为电子类各专业本科生、研究生的教材和相关领域工程技术人员的参考书, 也可作为本科 EDA 技术课程的后续课程教材和现代电子系统设计、电子设计竞赛、数字通信系统以及 Nios II 嵌入式系统高层次开发的参考书。

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

基于 FPGA 的嵌入式开发与应用/徐光辉, 程东旭, 黄如等编著. —北京: 电子工业出版社, 2006.9
(EDA 工具应用丛书)

ISBN 7-121-03073-X

I. 基… II. ①徐… ②程… ③黄… III. 可编程序逻辑器件—系统设计 IV. TP332.1

中国版本图书馆 CIP 数据核字 (2006) 第 095137 号

责任编辑: 雷洪勤

印 刷: 北京民族印刷厂

装 订: 北京鼎盛东极装订有限公司

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

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 27.5 字数: 700 千字

印 次: 2006 年 9 月第 1 次印刷

印 数: 5 000 册 定价: 43.00 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, 联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn。

服务热线: (010) 88258888。

Preface

FPGAs create an ideal platform for innovation in applications ranging from video broadcast systems and WiMAX basestations to networking equipment and automotive telematics. They combine flexibility, cost, and time-to-market advantages with high performance, system integration, and state-of-the-art development tools. As Moore's law continues to drive up semiconductor density while driving down costs, FPGAs are no longer reserved just for established companies, but are now accessible to engineering students around the world. This provides students the perfect learning environment for digital design today, using the very latest and most advanced software tools and techniques currently employed by experienced design engineers. Students can create both simple and sophisticated systems, gaining technical experience that makes them highly valuable to prospective employers.

Altera is committed to bringing the flexibility of programmable logic to students worldwide. We continue to increase our investments in China, creating training laboratories in top schools throughout the country to help professors and students develop FPGA design expertise. Young men and women studying computer science and electrical engineering today in China will be the architects and designers of next-generation systems. These engineers will drive China's continued economic growth.

Professor Xu's book is an excellent example of the partnership Altera fosters with the educational community. By helping students understand the fundamentals of FPGA technology and how it can be applied, including a number of real-life examples, this book fosters innovation while providing practical skills for their future. With the knowledge students gain from this book, there is no limit to what they can create.

Keep the fires of innovation burning!

Ben Lee
Vice President, Altera Asia Pacific

序

FPGA 为创新设计提供了一个理想的平台。从视频广播系统、WiMAX 基站，到网络设备和汽车远程信息处理，FPGA 的应用无所不在。FPGA 综合了灵活性、低成本和快速上市周期的优势，同时还具备高性能、系统集成和最先进的开发工具。摩尔关于半导体密度提高同时成本下降的预言还在继续，FPGA 已不再是大公司的专利，全世界各地的学生也能伸手可及。这为今天的学生们提供了一个完美的数字设计的学习环境。学生们可以使用最富经验的设计工程师正使用的最新和最先进的软件工具与技术，构建简单的或复杂的系统，积累技术经验，从而提高自己在以后工作岗位上的价值。

Altera 致力于为世界范围的学生们带来灵活的可编程逻辑。我们继续增加我们在中国的投资，在全国各地的高校内创建培训实验室，以帮助老师和学生们使用 FPGA 进行开发和设计。今天中国学习计算机科学和电子工程专业的年青人，将是下一代系统的建筑师和设计师。这些工程师将带动中国经济持续增长。

徐教授的书是 Altera 与教育界合作的一个优秀范例。通过帮助学生们理解 FPGA 技术的基本原理以及如何应用，包括一系列现实生活的实例，本书在为将来提供实用技巧的同时培养了学生们的创新意识。学生们通过从本书获取的知识，将获得无穷的创造力。

让创新之火永不熄灭！

李 彬

Altera 亚太区副总裁

前 言

随着电子技术的不断发展和进步,电子系统的设计方法发生了很大的变化,基于 EDA 技术的芯片设计正在成为电子系统设计的主流。集成电路向着高速度、高集成度和低功耗的系统集成方向迅猛发展,在单芯片上集成了嵌入式 CPU、DSP、存储器和其他各种接口的片上系统(SOC)正处于高速发展中。SOC 以嵌入式系统为核心,集软件硬件为一体,并追求产品系统最大包容的集成,是微电子领域 IP 设计的必然趋势和最终目标。

SOPC (System On Programmable Chip) 是 Altera 公司提出的片上可编程系统解决方案,它将 CPU、存储器、I/O 接口、DSP 模块、低电压差分信号(LVDS)技术、时钟数据恢复技术(CDR)以及锁相环(PLL)等系统设计所必需的模块集成到一片 FPGA 上,构成一个可编程的片上系统,使所设计的电路系统在其规模、可靠性、体积、功耗、功能、上市周期、开发成本、产品维护及硬件升级等多方面实现最优化。

SOPC 在应用理论和知识构成上,达到了一种有机融合。由于同时涉及底层的硬件系统和相应的软件设计,SOPC 技术使开发者能在软硬件系统的综合与构建两个方面充分发挥自己的创造性和想像力,从而多角度、多因素和多结构层面上对自己的设计进行优化。SOPC 从设计层次上讲,分硬件设计和软件设计;从设计流程上说,是典型的自顶向下的流程;从设计手段上,它更广泛和深入地利用计算机这一当今科研开发的主流技术。在这一平台上,设计者可以最大限度地优化系统的性能。从未来的电子系统设计趋势上看,SOPC 技术更具有发展性和前瞻性。

Altera 公司在 2000 年开始推出的第一代 Nios 软核处理器获得成功的基础上,于 2004 年 5 月正式推出第二代 Nios II 软核处理器,其性能超过 200 DMIPS,最低成本却只需 35 美分。与此同时,Altera 还推出了功能完善的开发套件,包括 C/C++编译器、集成开发环境(IDE)、JTAG 调试器、实时操作系统(RTOS)和 TCP/IP 协议栈。这些开发套件,配合 Quartus II 开发软件中的 SOPC Builder 设计工具,以及基于 Simulink 的系统级设计模块 DSP Builder,设计者可以很快完成一个 SOPC 系统的设计工作。

众所周知,信号与信号处理是信息科学中近十几年来发展最为迅速的学科之一。而 FPGA 正处于革命性数字信号处理的前沿。全新的 FPGA 系列正在越来越多地替代 ASIC 和 PDSP 用做数字前端数字信号处理的运算。正如我们现在已经看到的,随着在数字信号处理中的大规模应用,FPGA 正在日渐深入地进入我们的生产和生活,也必将在这一领域引起深刻的变革。

正因如此,本书在介绍 Nios II 嵌入式处理器开发与应用的同时,精选了大量 FPGA 在通信、数字信号处理及加密算法等领域的应用,这些实例都是作者对科研工作的总结。事实上,只有将 FPGA 应用到数字信号处理中,Nios II 的低成本才能真正体现出来。

本书以 Altera 公司的产品为背景,系统介绍 CPLD 和 FPGA 典型产品的结构原理、性能、特点、设计方法以及相应的 EDA 工具软件,重点介绍了 FPGA 在数字系统设计、数字通信与数字信号处理等领域中的应用,对 Nios II 嵌入式微处理器进行了深入的剖析,包括处理器架构、Avalon 总线、外设编程以及应用实例。书的最后还给出了 8051 IP 核和 PicoBlaze IP 核在 Altera FPGA 上的性能测试,这将对读者更为全面地理解 Nios II 处理器的性能有很大的帮助。

本书在选材上注重内容新颖、技术先进,并在书中给出了经实践验证的大量设计实例,

希望能对读者快速掌握基于 FPGA 的嵌入式系统设计与应用有所帮助。

本书由徐光辉、程东旭、黄如和黄智宇共同编写。徐光辉副教授担任大部分编写工作，程东旭工程师编写了第 4、9 章部分章节，黄如博士编写了第 4、7 章部分章节，黄智宇博士参与了第 1 章的编写。全书由徐志军教授统稿并审校。

本书的撰写得到了 Altera 公司大学计划部经理 Bob Xu 先生的大力支持，在此表示衷心的感谢。特别感谢 Altera 公司副总裁 Ben Lee 先生百忙中抽时间为本书作序。感谢实验室的吴浩涵、邱佳飞、周顺、郭锐和徐玉锋等硕士研究生们。

SOPC 技术是发展的，内容是变化的，涉及面也在不断扩充。由于编著者水平有限，加之时间仓促，书中难免有错误和不妥之处，恳请读者批评指正。

作 者

目 录

第 1 章 绪论	(1)
1.1 嵌入式系统简介	(1)
1.1.1 嵌入式系统的发展简介	(1)
1.1.2 嵌入式系统的概念与组成	(2)
1.1.3 嵌入式系统的特点	(3)
1.1.4 嵌入式系统的应用模式与发展趋势	(5)
1.2 SOPC 技术简介	(6)
1.2.1 SOPC 技术的主要特点	(7)
1.2.2 SOPC 技术实现方式	(8)
1.2.3 SOPC 系统开发流程	(9)
第 2 章 Altera 可编程逻辑器件简介	(10)
2.1 MAX II 器件	(10)
2.1.1 成本优化的架构	(10)
2.1.2 低功耗	(11)
2.1.3 高性能	(11)
2.1.4 用户 Flash 存储器	(12)
2.1.5 实时系统可编程能力 (ISP)	(13)
2.1.6 灵活的多电压 Multivolt 内核	(13)
2.1.7 JTAG 翻译器	(14)
2.1.8 I/O 能力	(15)
2.2 Cyclone 器件	(16)
2.2.1 新型可编程架构	(16)
2.2.2 嵌入式存储资源	(17)
2.2.3 专用外部存储接口电路	(18)
2.2.4 支持的接口及协议	(20)
2.2.5 锁相环的实现	(21)
2.2.6 I/O 特性	(22)
2.2.7 Nios II 嵌入式处理器	(23)
2.2.8 配置方案	(24)
2.3 Cyclone II 器件	(24)
2.3.1 主要特性	(24)
2.3.2 数字信号处理应用	(25)
2.3.3 专用外部存储器接口	(26)
2.3.4 嵌入式锁相环	(27)
2.3.5 单端 I/O 特性	(28)
2.3.6 差分 I/O 特性	(29)

2.3.7	自动 CRC 检测	(30)
2.3.8	Nios II 嵌入式处理器	(30)
2.4	Stratix 器件	(30)
2.4.1	高性能架构加快模块化设计	(31)
2.4.2	TriMatrix 存储器	(32)
2.4.3	DSP 块	(33)
2.4.4	高带宽 I/O 标准和高速接口	(34)
2.4.5	用于系统时钟管理的 PLL	(36)
2.4.6	器件配置和远程系统升级	(37)
2.5	Stratix II 器件	(38)
2.5.1	新型逻辑结构	(38)
2.5.2	高速 I/O 信号和接口	(40)
2.5.3	外部存储器接口	(41)
2.5.4	针对 Stratix II 器件优化的 IP	(43)
2.5.5	设计安全性	(43)
2.5.6	TriMatrix 存储器	(43)
2.5.7	数字信号处理块	(44)
2.5.8	时钟管理电路	(46)
2.5.9	片内匹配	(47)
2.5.10	远程系统升级	(48)
2.6	Stratix GX 器件	(48)
第 3 章	Quartus II 软件应用	(51)
3.1	图形用户界面设计流程	(51)
3.2	命令行设计流程	(53)
3.3	交通灯设计实例	(54)
3.3.1	设计原理	(55)
3.3.2	设计输入	(56)
3.3.3	创建工程	(57)
3.3.4	编译前设置	(61)
3.3.5	编译	(63)
3.3.6	仿真	(64)
3.3.7	应用 RTL 电路图观察器	(67)
3.4	引脚锁定和下载验证	(68)
3.4.1	引脚锁定	(68)
3.4.2	下载验证	(70)
3.4.3	对配置器件编程	(71)
3.5	使用嵌入式逻辑分析仪进行实时测试	(72)
3.5.1	SignalTap II 逻辑分析仪使用流程	(72)
3.5.2	编译特定逻辑条件触发信号	(76)
3.6	使用在系统嵌入式存储器	(77)
3.6.1	正弦信号发生器的设计	(78)

3.6.2	定制ROM 初始化数据文件	(79)
3.6.3	定制ROM 元件	(80)
3.6.4	使用在系统嵌入式存储器数据编辑器	(83)
3.7	嵌入式锁相环 altPLL 宏功能模块调用	(84)
	思考题	(87)
第4章	基于FPGA的DSP算法实现	(88)
4.1	CORDIC 算法的FPGA实现	(88)
4.1.1	CORDIC 原理	(88)
4.1.2	CORDIC 处理器的预处理与后处理	(90)
4.1.3	CORDIC 算法的FPGA实现	(91)
4.1.4	迭代结构CORDIC 算法的VHDL实现	(93)
4.1.5	流水线结构CORDIC 算法的VHDL实现	(96)
4.2	CRC 码的FPGA实现	(102)
4.2.1	CRC 校验的原理	(102)
4.2.2	CRC 码的Verilog 编程实现	(104)
4.2.3	CRC 码的仿真验证和逻辑综合	(108)
4.3	Viterbi 译码器的FPGA实现	(109)
4.3.1	Viterbi 译码器的系统结构划分	(109)
4.3.2	Viterbi 译码器的Verilog 编程实现	(112)
4.4	DES 算法的可编程硬件实现	(132)
4.4.1	DES 算法的数据置换	(134)
4.4.2	DES 算法的密钥置换	(135)
4.4.3	DES 的解密过程	(137)
4.4.4	DES 的Verilog 编程实现及其逻辑综合	(137)
4.5	SHA-1 算法的可编程硬件实现	(144)
4.5.1	SHA-1 算法描述	(144)
4.5.2	SHA-1 的Verilog 编程实现及其仿真测试	(145)
4.6	ECCDSA 算法的可编程硬件实现	(156)
4.6.1	ECCDSA 算法描述	(156)
4.6.2	ECCDSA 算法的Verilog 编程实现	(158)
4.6.3	ECCDSA 算法的仿真验证和逻辑综合	(190)
第5章	Nios II 处理器结构	(191)
5.1	概 述	(191)
5.2	Nios II 系统的特性	(192)
5.3	Nios II 处理器构架简介	(193)
5.3.1	寄存器文件	(193)
5.3.2	ALU	(193)
5.3.3	异常控制器和中断控制器	(194)
5.3.4	存储器和I/O 组织	(194)
5.4	编程模型	(198)
5.4.1	通用寄存器	(198)

5.4.2	控制寄存器	(199)
5.4.3	操作模式	(200)
5.4.4	异常处理	(201)
5.4.5	跳出处理	(204)
5.4.6	访问存储器和外设	(205)
5.4.7	处理器复位时的状态	(205)
5.4.8	Nios II 处理器的指令集	(206)
思考题		(209)
第 6 章 Avalon 总线规范		(210)
6.1	概述	(210)
6.1.1	Avalon 总线的特点	(210)
6.1.2	术语和概念	(211)
6.2	Avalon 总线信号	(213)
6.2.1	Avalon 信号类型详细列表	(213)
6.2.2	Avalon 信号时序说明	(216)
6.2.3	总线传输特性	(217)
6.3	Avalon 从端口传输	(217)
6.3.1	从端口信号分析	(217)
6.3.2	基本从端口读传输	(219)
6.3.3	具有固定等待周期的从端口读传输	(220)
6.3.4	具有可变等待周期的从端口读传输	(221)
6.3.5	具有建立时间的从端口读传输	(222)
6.3.6	基本从端口写传输	(223)
6.3.7	具有固定等待周期的从端口写传输	(224)
6.3.8	具有可变等待周期的从端口写传输	(224)
6.3.9	具有建立时间和保持时间的从端口写传输	(225)
6.4	Avalon 主端口传输	(226)
6.4.1	主端口信号分析	(226)
6.4.2	基本主端口读传输	(228)
6.4.3	基本主端口写传输	(229)
6.5	流水线传输	(230)
6.5.1	带固定延迟的流水线从端口读传输	(231)
6.5.2	带可变延迟的流水线从端口读传输	(232)
6.5.3	流水线主端口读传输	(234)
6.6	流传输模式	(235)
6.6.1	流传输模式信号分析	(235)
6.6.2	流传输模式从端口读传输	(236)
6.6.3	流传输模式从端口写传输	(237)
6.6.4	流传输模式主端口传输	(238)
6.7	三态传输	(240)
6.7.1	三态从端口传输介绍	(240)

6.7.2 三态从端口传输实例	(243)
6.7.3 三态主端口传输	(245)
6.8 突发传输	(246)
6.8.1 主端口突发写传输	(247)
6.8.2 主端口突发读传输	(248)
6.8.3 从端口突发写传输	(250)
6.8.4 从端口突发读传输	(251)
6.9 非传输相关信号	(252)
6.10 地址对齐	(253)
6.10.1 本地地址对齐	(253)
6.10.2 动态地址对齐	(254)
思考题	(255)
第 7 章 基于 SOPC 的 Nios II 处理器设计	(256)
7.1 SOPC Builder 功能	(256)
7.2 SOPC Builder 组成	(258)
7.2.1 系统内容页面	(259)
7.2.2 Board 设置页面	(260)
7.2.3 CPU 设置页面	(260)
7.2.4 SDRAM 设置页面	(261)
7.2.5 系统生成页面	(262)
7.3 SOPC Builder 组件	(263)
7.3.1 库搜索路径	(263)
7.3.2 class.ptf 文件	(263)
7.3.3 cb_generator.pl 文件	(264)
7.3.4 系统 PTF 文件	(264)
7.4 SOPC Builder 应用实例	(265)
7.4.1 使用 SOPC Builder 创建 Nios 系统	(265)
7.4.2 将 Nios 处理器加入 Quartus II 项目	(270)
7.4.3 在 Nios II IDE 下运行 Hello world 程序	(276)
7.5 自定义逻辑	(279)
7.5.1 组件编辑器	(279)
7.5.2 组件开发流程	(280)
7.5.3 自定义 PWM 组件实例	(282)
7.5.4 小结	(290)
7.6 定制目标板程序	(291)
7.6.1 目标板程序	(291)
7.6.2 使用 mk_target_board	(291)
7.6.3 在 SOPC Builder 中指定目标板	(295)
7.6.4 使用 Flash Programmer	(296)
7.6.5 小结	(298)
思考题	(299)

第 8 章 Nios II 外设及其编程	(300)
8.1 并行输入输出 (PIO)	(300)
8.1.1 功能描述	(300)
8.1.2 寄存器映射	(301)
8.1.3 软件编程	(303)
8.1.4 PIO 实例分析	(304)
8.2 通用异步收发器 (UART)	(306)
8.2.1 功能描述	(306)
8.2.2 在 SOPC Builder 中添加 UART	(308)
8.2.3 软件编程模型	(309)
8.3 定时器	(310)
8.3.1 功能描述	(310)
8.3.2 在 SOPC Builder 中添加定时器	(311)
8.3.3 软件编程模型	(313)
8.4 JTAG UART	(315)
8.4.1 功能描述	(315)
8.4.2 在 SOPC Builder 中添加 JTAG UART	(317)
8.4.3 软件编程模型	(318)
8.5 SPI	(320)
8.5.1 功能描述	(320)
8.5.2 在 SOPC Builder 中添加 JTAG UART	(323)
8.5.3 软件编程模型	(324)
8.6 系统 ID	(327)
8.6.1 功能描述	(327)
8.6.2 在 SOPC Builder 中添加系统 ID	(327)
8.6.3 软件编程模型	(327)
8.7 字符 LCD 控制器	(328)
8.7.1 功能描述	(328)
8.7.2 在 SOPC Builder 中添加 LCD 控制器	(328)
8.7.3 软件编程模型	(328)
8.8 数据选择器	(329)
8.8.1 功能描述	(329)
8.8.2 在 SOPC Builder 中添加 JTAG UART	(330)
8.8.3 软件编程模型	(330)
8.9 邮箱	(331)
8.9.1 功能描述	(331)
8.9.2 在 SOPC Builder 中添加邮箱	(331)
8.9.3 软件编程模型	(332)
8.10 SDRAM 控制器	(332)
8.10.1 功能描述	(333)
8.10.2 在 SOPC Builder 中添加 SDRAM 控制器	(335)

8.10.3	硬件仿真	(336)
8.10.4	SDRAM 控制器配置实例	(337)
8.10.5	软件编程模型	(338)
8.10.6	SDRAM 编程实例	(338)
8.11	CFI 控制器	(340)
8.11.1	功能描述	(340)
8.11.2	在 SOPC Builder 中添加 CFI 控制器	(340)
8.11.3	软件编程模型	(341)
8.11.4	Flash 器件编程实例	(341)
8.12	EPCS 器件控制器	(342)
8.12.1	功能描述	(343)
8.12.2	在 SOPC Builder 中添加 EPCS 控制器	(344)
8.12.3	软件编程模型	(344)
第 9 章	嵌入式处理器应用实例	(345)
9.1	基于 DM9000A 的 ARP 协议实现	(345)
9.1.1	ARP 协议的工作原理	(345)
9.1.2	DM9000A 工作原理	(347)
9.1.3	SOPC Builder 下 DM9000A 的自定义逻辑	(349)
9.1.4	ARP 协议的实现	(351)
9.1.5	小结	(359)
9.2	基于 HD61202 的 128 × 64 液晶显示模块的应用	(360)
9.2.1	ATM12864 的性能特点	(360)
9.2.2	HD61202 的性能特点	(360)
9.2.3	SOPC Builder 下 HD61202 的自定义逻辑	(363)
9.2.4	LCD 汉显的实现	(365)
9.3	MC8051 IP 核的应用开发	(369)
9.3.1	MC8051 IP 核的结构分析	(371)
9.3.2	MC8051 IP 核的逻辑综合	(380)
9.3.3	MC8051 IP 核的功能仿真	(381)
9.4	PicoBlaze IP 核在 ALTERA 公司 FPGA 上的性能测试	(388)
9.4.1	CPLD 型 PicoBlaze 处理器基本结构	(388)
9.4.2	PicoBlaze 处理器的指令系统	(390)
9.4.3	PicoBlaze 处理器的应用测试	(394)
附录 A	VHDL 基本语法	(400)
A.1	程序结构	(400)
A.1.1	实体	(400)
A.1.2	ARCHITECTURE 结构体	(401)
A.1.3	LIBRARY 库	(402)
A.1.4	PACKAGE 程序包	(402)
A.1.5	CONFIGURATION 配置	(403)
A.2	语言要素	(403)

A.2.1	VHDL 文字规则	(403)
A.2.2	数据对象	(405)
A.2.3	数据类型	(405)
A.2.4	操作符	(406)
A.3	顺序语句	(408)
A.3.1	赋值语句	(408)
A.3.2	流程控制语句	(408)
A.3.3	子程序调用语句	(410)
A.4	并行语句	(411)
A.4.1	进程语句	(411)
A.4.2	块语句	(412)
A.4.3	元件例化语句	(412)
附录 B	Verilog HDL 基本语法	(414)
B.1	词法约定	(414)
B.1.1	空白符和注释符	(414)
B.1.2	操作符	(414)
B.1.3	数字声明	(414)
B.1.4	字符串	(415)
B.1.5	标识符和关键字	(415)
B.2	数据类型	(415)
B.2.1	物理数据类型	(415)
B.2.2	抽象数据类型	(416)
B.3	运算符	(416)
B.3.1	运算符的类型	(416)
B.3.2	运算符的优先级	(418)
B.4	模块和端口	(418)
B.4.1	模块的组成	(418)
B.4.2	模块的端口定义	(419)
B.4.3	模块的调用	(420)
B.5	行为描述	(421)
B.5.1	过程语句	(421)
B.5.2	过程赋值语句	(422)
B.5.3	分支语句	(422)
B.5.4	循环语句	(423)

第1章 绪 论

1.1 嵌入式系统简介

电子数字计算机诞生于 1946 年,在此后很长一段时间之内,我们都按照计算机的体系结构、运算速度、结构规模和适用领域等,将其分成大型计算机、中型计算机、小型计算机和微计算机,并以此来组织学科和产业分工,这种分类沿袭了约 40 年。近 10 年来,随着电子技术和计算科学的快速发展,特别是超大规模集成电路的发展,为科学技术研究带来了惊人的推动力,使得实际情况产生了根本性的变化,例如 20 世纪 70 年代末出现的由微计算机演变出来的个人计算机(PC),如今已经占据了全球计算机工业 90% 的市场,其处理速度也超过了当年的大、中型计算机。随着计算机技术和产品对其他行业的广泛渗透,以应用为中心的分类方法变得更为切合实际,也就是按计算机的嵌入式应用和非嵌入式应用将其分为嵌入式计算机和通用计算机。

通用计算机具有计算机的标准形态,通过装配不同的应用软件,以通用的形式出现并应用在社会各个方面,其典型产品为 PC;而嵌入式计算机则是以嵌入式系统的形式隐藏在各种装置、产品和系统中。嵌入式控制技术的出现为工业控制领域带来了一场新的技术革命。由嵌入式微控制器所组成的系统,不仅在大型设备中应用广泛,其更大的优势是可以嵌入到微型或小型仪器中。嵌入式计算机在应用数量上远远超过了各种通用计算机,一台通用计算机的外部设备中就包含了 5~10 个嵌入式微处理器,键盘、鼠标、软驱、硬盘、显示卡、显示器、Modem、网卡、声卡、打印机、扫描仪、数码相机、USB 集线器等都是由嵌入式处理器控制的。在制造业、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等方面均是嵌入式计算机的应用领域。

21 世纪,在全球新一轮汽车、通信、信息电器、医疗、军事等行业的巨大智能化装备市场需求下,全球嵌入式软件及系统产业得到了快速发展,可以说已经广泛地应用于人类生活的方方面面。在中国,虽然嵌入式操作系统行业刚刚进入起步阶段,但进展神速。2004 年我国嵌入式系统应用产品经济总量估计超过 10000 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为 13 亿片。嵌入式处理器的应用已无处不在,其市场规模已接近通用计算机的两倍,且增长速度强劲。其应用主要包括 DVD 播放器、机顶盒、音视频接收设备、MP3 播放器、数码相机和汽车电子等各个领域。发展嵌入式系统产业成为我国信息产业由“中国制造”向“中国创造”的突破口,成为我国信息产业增长方式由粗放型向集约型转变,实现可持续发展的重要途径。

1.1.1 嵌入式系统的发展简介

在电子计算机诞生的初期,它始终是“供养”在特殊的机房中,实现数值计算的大型昂贵设备。直到 20 世纪 70 年代,随着微处理器的出现,计算机才出现了历史性的变化。以微处理器为核心的微型计算机以其小型、价廉、高可靠性特点,迅速走出机房。同时,基于高速数值

计算能力的微型机也为控制领域的发展带来了极大的契机。

由于嵌入式计算机系统要嵌入到对象体系中,实现的是对象的智能化控制,因此,它有着与通用计算机系统完全不同的技术要求和技術发展方向。

通用计算机系统的技术要求是高速、海量的数值计算,技術发展方向是总线速度的无限提升,存储容量的无限扩大。计算机专业领域集中精力发展通用计算机系统的软、硬件技术,通用微处理器迅速从 286、386、486 到如今的奔腾系列;操作系统则迅速扩张计算机基于高速海量的数据文件处理能力,使通用计算机系统进入到尽善尽美的阶段;而嵌入式计算机系统的技术要求则是对对象的智能化控制能力,技術发展方向是与对象系统密切相关的嵌入性能、控制能力与控制的可靠性。早期,人们勉为其难地将通用计算机系统进行改装,在大型设备中实现嵌入式应用。然而,对于众多的对象系统(如家用电器、仪器仪表、工控单元……),无法嵌入通用计算机系统,况且嵌入式系统与通用计算机系统的技術发展方向完全不同,因此,必须独立地发展通用计算机系统与嵌入式计算机系统。嵌入式计算机系统动员了原有的传统电子系统领域的厂家与专业人员,迅速地将传统的电子系统发展到智能化的现代电子系统时代。

嵌入式系统真正兴起是在 20 世纪 60 年代末,当时的计算机技术处于以电路集成化为标志第三代时期。由于军事和空间领域的需求推动,计算机的软硬件技术达到了可以把人送上月球再返回地面的可靠性要求,而且在这一时期计算机技术取得重大发展,如并行技术、先行控制、流水线、单总线结构、高速通用寄存器、强有力的中断系统、操作系统等。

嵌入式系统的真正发展是在微处理器问世之后。1971 年 11 月,Intel 公司成功地把算术运算器和控制器电路集成在一起,推出了世界上第一片微处理器 Intel 4004。从此,嵌入式处理器经历了从简单到复杂,从 4 位处理器到 64 位处理器的发展过程。最初 4 位单片机的处理能力有限,而后来推出的 8 位单片机由于运算速度提高、处理能力增强且编程设计开发方便,因此得到迅速推广,到目前仍然有广泛应用。8 位单片机的典型代表有早期流行的 CISC 结构的 Z80 单片机、MCS51 单片机和目前得到广泛应用的 RISC 结构的 AVR 单片机、PIC 单片机。16 位处理器典型产品有 Intel 8086/80186、Z8000、M680000、DSP 等。16/32 位嵌入式处理器进一步提高了处理速度、扩大了可支配资源等,但成本相对较高,一般用于较高端领域。

1.1.2 嵌入式系统的概念与组成

在计算机领域中,对于嵌入式系统的概念,至今没有统一严格的定义。如果我们了解嵌入式计算机的由来与发展,就不会对嵌入式系统产生过多的误解,就能历史地、本质地、普遍适用地定义嵌入式系统。

“嵌入式计算机系统是指嵌入到对象体系中的专用计算机系统”。“嵌入性”、“专用性”与“计算机系统”是嵌入式系统的三个基本要素。嵌入式系统的软、硬件必须可裁剪,它是满足对象要求的最小软、硬件配置。由于是嵌入到对象系统(对象系统是指嵌入式系统所嵌入的宿主系统)中,必须满足对象系统的环境要求,如物理环境(小型)、电气/气氛环境(可靠)、成本(价廉)等要求。嵌入式系统与对象系统密切相关,其主要技术发展方向是满足嵌入式应用要求,不断扩展对象系统要求的外围电路(如 ADC、DAC、PWM、日历时钟、电源监测、程序运行监测电路等),形成满足对象系统要求的应用系统。因此,嵌入式系统作为一个专用的计算机系统,要不断向计算机应用系统发展。因此,可以把定义中的专用计算机系统引伸成:满足对象系统要求的计算机应用系统。