# Nios II 嵌入式软核 SOPC 设计原理及应用

李兰英　等编著

北京航空航天大学出版社

# 前　言

Altera 公司 Nios II 软核概念的提出及 SOPC 的软硬件综合解决方案，彻底颠覆了传统的嵌入式系统的设计理念，从硬件和软件整体设计上将嵌入式系统设计进行了极大的推动，使得嵌入式系统的硬件电路更加简单、有效，易于理解；软件设计变得轻松，移植性更强。

与传统的仅采用基于微处理器的软件设计或者采用 ASIC/FPGA 的硬件实现相比，SOPC 的软硬件协同设计方法完全不同。它可根据应用系统的不同要求，适当地划分软硬件的功能，以求得最佳的性能价格比；同时不需要预先制作实验电路板，这可节省大量的人力、物力；另外，SOPC 技术使开发者在软硬件系统的综合与构建方面可以充分发挥自己的创造性和想象力，对系统进行优化。

Altera 公司提供了完整的开发套件：Quartus II、SOPC Builder、Nios II IDE 和仿真工具等软件，并将其无缝地集成在一起。这为当前嵌入式系统设计提供了新方法，并带来了巨大的便利。

较之传统的硬核处理器设计，SOPC Builder 是基于 SOC 和 IP 的思想进行设计的，采用了很多参数化的 IP，隐藏了很多初始化的细节，使得硬件开发相当容易；同时它还提供完备的 C 语言头文件，隐藏了很多硬件细节，降低了软件的开发难度，外设丰富，集成简单；除了它本身提供大量基于 Avalon 总线的 IP 外，它还是一个开放的 IP 集成环境。用户可以很容易地将自己设计的 IP 集成到 SOPC Builder 中，实现设计重用。

Nios II 集成开发环境(IDE)是 Nios II 系列嵌入式处理器的基本软件开发工具。所有软件开发任务都可以在 Nios II IDE 下完成，包括编辑、编译、程序调试、下载和运行。Nios II IDE 具有编码生成环境以及可选 RTOS 和 TCP/IP 库集成。它还提供构建管理工具，使用 GNU 编译器作为其支撑技术。此外，Nios II IDE 还具有快闪编程器功能。

Nios II 嵌入式处理器是一种面向用户的、可以灵活定制的通用 RISC(精简指令集架构)嵌入式 CPU。它是一款具有广阔应用前景的处理器，融入了许多新的设计方法和理念，其 SOPC 概念体现在以下几个方面：

- Nios II 符合工业技术的发展潮流，即硬件设计软件化。采用 Nios II 能有效地降低人力和物力成本，提高产品竞争力；硬件设计软件化还能方便对硬件进行仿真、验证，整体系统结构的数字逻辑化设计使得验证工作可通过仿真软件顺利地实现，可以掌握详细、清楚的信息；减少了硬件设计的错误，使得对硬件接口不是很熟悉的人也可以进行系统平台的集成。

- 采用 Nios II 可以提高系统的鲁棒性，这也是单芯片解决方案带来的好处。Nios II 软核及其开发平台帮助开发者将大部分模块构建好，却又不失灵活性，对大多数外设开发了相应的驱动程序。这无疑使得人为的设计失误率降到最低，硬件设计难度降低，开发周期缩短。
- Nios II 帮助用户保护自己的知识产权，能够在很大程度上杜绝仿制。
- Nios II 的价值在于它为 FPGA 的应用拓展了新的方向，真正在 FPGA 上实现了 SOPC。

随着相关技术的不断发展，相信不久的将来，所有的系统设计者都可以共享真正的 SOPC 解决方案，通过先进的工具，用最简单的方式设计最复杂的系统。

选用 Altera 的基于 Nios II 的 SOPC 平台来进行系统设计，非常适合业界及院校的教学、实验及快速的产品开发。

要深入学习 SOPC 系统的开发技术，要求设计者必须了解基本的 EDA 软件、硬件描述语言和 FPGA 器件相关知识，掌握计算机组成与接口、汇编语言和 C 语言、DSP 算法、嵌入式系统开发及嵌入式操作系统等知识。

SOPC 技术涉及面很广，作为在嵌入式系统开发中另一条可选择的途径，本书只能起到抛砖引玉的作用。本教程的编写目的是期望初学者少走弯路，迅速掌握 SOPC 的设计理念和方法，逐步提高嵌入式系统的设计和开发能力。

全书共分 10 章，各章主要内容如下：

第 1 章　概述 SOPC 及其相关概念、系统组成、开发环境、FPGA 的基础知识，最后对书中示例所使用的 FPGA 器件 Cyclone II EP2C35 作了简要介绍。

第 2 章　介绍 Nios II 处理器软核的体系结构。

第 3 章　详细介绍 Avalon 总线的概念、组成、传输模式、信号类型。

第 4 章　详细介绍 SOPC Builder 中的常用外设接口 IP 核的功能、参数配置、编程模型。包括 SDRAM 控制器、通用 Flash 接口、EPCS 控制器、PIO、定时器、UART、JTAG UART 核、SPI、DMA、系统 ID、PLL 核及 Mutex 核等。

第 5 章　通过一个简单实例，详细讨论使用 Quartus II 和 SOPC Builder 软件开发硬件系统的流程和方法。

第 6 章　通过一个简单实例，说明基于 Nios II IDE 的应用软件开发流程和方法，并对其进行了系统仿真。

第 7 章　对 FPGA 配置和编程技术进行了介绍，包括配置芯片、配置模式、配置器件的信号连接及配置数据的编程方法等内容。

第 8 章　对基于 Nios II 的 SOPC 的高级技术进行深入探讨，包括异常处理程序的开发、缓存和紧耦合存储器的编程、基于 μC/OS - II 的实时操作系统的应用程序设

计、以太网与轻量 IP、Nios II 多处理器系统、定制指令的设计以及定制用户 IP 的设计与实现。

第 9 章　通过一个开发实例，详细说明 SOPC 应用系统的设计与实现方法。

第 10 章　对 Quartus II 中集成的仿真工具 Simulator、SignalTap II 逻辑分析仪及第三方仿真软件 ModelSim 的功能、使用方法及在 SOPC 中的应用进行了阐述。

本书特点：

- 思路清晰　本书从教学的角度，在章节安排顺序和内容上延续微机原理、单片机、嵌入式系统的教学思路，章节安排围绕 SOPC 硬件系统设计和 SOPC 应用程序开发两条主线展开，条理清晰，便于学生快速理解基于 Nios II 嵌入式软核 SOPC 应用系统设计技术与单片机应用系统、固核嵌入式系统的相同点与不同点。
- 全面系统　本书对基于 Nios II 嵌入式软核的 SOPC 软硬件设计环境、原理、方法及应用进行了全面、系统、详细的论述，内容由浅入深，既适合 SOPC 技术的初学者，又适合开发 SOPC 复杂应用系统的工程技术人员。
- 内容新颖　书中介绍的开发工具(Quartus II 5.1)及参考资料均为 Altera 公司的最新版本。
- 实践性强　作者均从事单片机、嵌入式理论和实验教学、科研多年，结合多年的教学经验与应用心得，从较深的层面对各种设计工具和设计方法进行研究，且通过大量具体的实例加以说明，所有结果均通过上机实验得到验证。

SOPC 代表了嵌入式系统的发展方向，我们有理由相信，不久的将来，SOPC 的身影就像今天的单片机一样随处可见。因此，掌握 SOPC 技术者必将成为引领未来嵌入式系统设计的先锋。本书的出版具有一定的前瞻性和发展性。

参加本书编写的有：李兰英(第 1、2、3、4、7 章)，崔永利(第 5、6 章及附录 A、B)，李霄燕(第 8 章)，李妍(第 10 章及附录 C)，董怀国、何云斌(合写第 9 章)。由李兰英担任主编并负责全书的统稿工作。

致谢：

在本书的编写过程中，得到了北京航空航天大学出版社的大力支持，也得到了我的同学张鑫、同事孟晓英、崔林海、洪国铭、陈为民等的大力帮助与关心。研究生冯宏伟、杨晨、高俊峰、曹望成、刘洋、张滇、张向国、夏云磊、刘小平、李丰旺、姜秀丽等同学亦参与了本书编写的部分工作，在此表示深深感谢！

我们真诚地希望读者对书中的不足和错误给予批评指正(E-mail：lulu08521@sina.com)。

作　者

2006.8

# 目　录


## 第 1 章　概　述

1.1　SOPC 及其技术 …… 1
1.1.1　嵌入式系统简介 …… 1
1.1.2　SOC 系统简介 …… 3
1.1.3　SOPC 技术简介 …… 5
1.1.4　CPLD/FPGA 简介 …… 9
1.1.5　新一代低成本 FPGA——Cyclone II 简介 …… 14
1.2　Nios II 软核 SOPC 系统及组件 …… 26
1.2.1　Nios II 软核嵌入式处理器 …… 29
1.2.2　Avalon Switch Fabric …… 30
1.2.3　外围设备 …… 32
1.3　Nios II 软核 SOPC 系统开发环境 …… 32
1.3.1　硬件开发环境 …… 32
1.3.2　软件开发环境 …… 33

## 第 2 章　Nios II 嵌入式软核处理器体系结构

2.1　Nios II 处理器的实现 …… 35
2.2　Nios II 的内部寄存器 …… 37
2.2.1　通用寄存器 …… 37
2.2.2　控制寄存器 …… 38
2.3　Nios II 的操作模式 …… 39
2.3.1　管理模式 …… 39
2.3.2　用户模式 …… 40
2.3.3　调试模式 …… 40
2.3.4　模式之间的切换 …… 40
2.4　Nios II 的异常处理 …… 41
2.4.1　硬件中断 …… 41
2.4.2　软件陷阱 …… 42

2.4.3 未实现指令 …… 42
2.4.4 确定异常产生的原因 …… 42
2.4.5 异常嵌套与返回 …… 43
2.4.6 断点处理与返回 …… 43
2.5 存储器与外设 …… 44
2.5.1 指令和数据总线 …… 45
2.5.2 高速缓存 …… 47
2.5.3 紧耦合存储器 …… 47
2.5.4 地址映射 …… 48
2.5.5 存储器与外设的访问 …… 48
2.6 处理器复位状态 …… 49
2.7 寻址方式与指令集分类 …… 49
2.7.1 寻址方式 …… 49
2.7.2 数据传送指令 …… 50
2.7.3 算术和逻辑运算指令 …… 50
2.7.4 移位和循环移位指令 …… 52
2.7.5 程序控制指令 …… 52
2.7.6 定制指令 …… 53
2.7.7 未实现指令 …… 53
2.8 JTAG 调试模块 …… 53

**第 3 章 Avalon 总线规范**

3.1 Avalon 总线概述 …… 57
3.1.1 功 能 …… 57
3.1.2 术语和概念 …… 58
3.2 Avalon 信号 …… 60
3.2.1 信号的列表 …… 61
3.2.2 信号的极性 …… 64
3.2.3 信号的命名 …… 64
3.2.4 信号的时序 …… 65
3.2.5 传输特性 …… 65
3.3 Avalon 从端口传输 …… 66
3.3.1 从端口信号详述 …… 66
3.3.2 从端口的读传输 …… 68

3.3.3 从端口的写传输 …… 73
3.4 Avalon 主端口传输 …… 77
3.4.1 主端口信号详述 …… 77
3.4.2 基本主端口的读传输 …… 78
3.4.3 基本主端口的写传输 …… 80
3.4.4 等待状态、建立和保持时间 …… 81
3.5 流水传输 …… 81
3.5.1 具有固定延迟的从端口流水读传输 …… 82
3.5.2 具有可变延迟的从端口流水读传输 …… 83
3.5.3 主端口流水读传输 …… 85
3.6 流传输模式 …… 87
3.6.1 流模式从端口传输 …… 87
3.6.2 流模式主端口传输 …… 91
3.7 三态传输 …… 92
3.7.1 三态从端口传输 …… 92
3.7.2 三态主端口传输 …… 99
3.8 突发传输 …… 100
3.8.1 限 制 …… 100
3.8.2 主端口突发传输 …… 101
3.8.3 从端口突发传输 …… 104
3.9 与传输无关的信号 …… 107
3.9.1 中断请求信号 …… 107
3.9.2 复位控制信号 …… 108
3.10 地址对齐 …… 109
3.10.1 静态地址对齐 …… 109
3.10.2 动态地址对齐 …… 110

**第 4 章 Nios II 外围设备**

4.1 SDRAM 控制器 …… 111
4.1.1 SDRAM 控制器核概述 …… 111
4.1.2 SOPC Builder 中的 SDRAM 控制器核配置选项 …… 114
4.1.3 配置实例 …… 117
4.1.4 SDRAM 软件编程模型 …… 118
4.1.5 时钟、PLL 和时序 …… 118

4.2 CFI 控制器 …… 122
4.2.1 CFI Flash 控制器核概述 …… 122
4.2.2 SOPC Builder 中 CFI 控制器核配置选项 …… 123
4.2.3 CFI 软件编程模型 …… 124
4.3 EPCS 设备控制器 …… 125
4.3.1 EPCS 控制器核综述 …… 125
4.3.2 SOPC Builder 中 EPCS 控制器核配置选项 …… 127
4.3.3 EPCS 软件编程模型 …… 127
4.4 PIO 控制器 …… 128
4.4.1 PIO 控制器核综述 …… 128
4.4.2 SOPC Builder 中 PIO 核配置选项 …… 130
4.4.3 PIO 软件编程模型 …… 131
4.4.4 PIO 寄存器描述与中断 …… 131
4.5 定时器控制器 …… 133
4.5.1 定时器控制器核综述 …… 133
4.5.2 SOPC Builder 中定时器核配置选项 …… 135
4.5.3 定时器软件编程模型 …… 136
4.5.4 定时器寄存器描述与中断 …… 137
4.6 UART 核 …… 139
4.6.1 UART 核综述 …… 139
4.6.2 SOPC Builder 中 UART 核配置选项 …… 142
4.6.3 UART 软件编程模型 …… 145
4.6.4 UART 寄存器描述与中断 …… 147
4.7 JTAG UART 核 …… 152
4.7.1 JTAG UART 核综述 …… 152
4.7.2 SOPC Builder 中 JTAG UART 核配置选项 …… 154
4.7.3 JTAG UART 的软件编程模型 …… 156
4.7.4 JTAG UART 寄存器描述与中断 …… 159
4.8 SPI 核 …… 161
4.8.1 SPI 核综述 …… 161
4.8.2 SPI 配置实例 …… 162
4.8.3 SOPC Builder 中 SPI 核配置选项 …… 164
4.8.4 SPI 软件编程模型 …… 166
4.8.5 SPI 寄存器描述 …… 167

4.9 DMA 控制器 …… 169
4.9.1 DMA 控制器核综述 …… 169
4.9.2 SOPC Builder 中 DMA 控制器核配置选项 …… 171
4.9.3 DMA 软件编程模型 …… 172
4.9.4 DMA 寄存器描述与中断 …… 173
4.10 系统 ID 核 …… 176
4.10.1 系统 ID 核综述 …… 176
4.10.2 SOPC 中系统 ID 核配置选项 …… 177
4.10.3 系统 ID 软件编程模型 …… 177
4.11 PLL 核 …… 177
4.11.1 PLL 核综述 …… 177
4.11.2 SOPC Builder 中 PLL 控制器核配置选项 …… 179
4.11.3 PLL 寄存器描述 …… 180
4.12 mutex 核 …… 182
4.12.1 概 述 …… 182
4.12.2 功能描述 …… 182
4.12.3 在 SOPC Builder 中使用 mutex 核 …… 183
4.12.4 mutex 软件编程模型 …… 183
4.12.5 mutex 核的 API …… 184

**第 5 章 简单 SOPC 硬件系统开发**

5.1 基于 Nios II 的 SOPC 硬件系统开发流程 …… 187
5.2 SOPC Builder 硬件开发环境介绍 …… 189
5.2.1 建立系统 …… 189
5.2.2 生成系统 …… 190
5.3 简单 SOPC 实例开发系统需求及任务 …… 190
5.4 创建 Quartus II 工程 …… 191
5.5 使用 SOPC Builder 创建 Nios II 系统模块 …… 194
5.5.1 启动 SOPC Builder 进行 Nios II 系统硬件设计 …… 195
5.5.2 设置目标 FPGA 及时钟 …… 196
5.5.3 添加 CPU 和 IP 模块 …… 198
5.5.4 指定基地址和中断 …… 213
5.5.5 配置 Nios II 系统 …… 214
5.5.6 生成 Nios II 系统 …… 216

5.6 集成 Nios II 系统到 Quartus II 工程 …… 217
5.6.1 创建顶层模块 …… 218
5.6.2 添加 Nios II 系统模块到 Quartus II 顶层模块 …… 218
5.6.3 添加引脚和其他基本单元 …… 218
5.6.4 命名引脚及引脚连接 …… 220
5.6.5 选择器件及分配 FPGA 引脚 …… 221
5.6.6 器件和引脚的其他设置 …… 224
5.7 Quartus II 工程编译 …… 227
5.7.1 创建编译器设置 …… 227
5.7.2 编译硬件系统 …… 228
5.8 编程下载 …… 229

**第 6 章 SOPC 软件开发**

6.1 SOPC 软件开发环境综述 …… 233
6.1.1 SOPC 软件开发流程 …… 234
6.1.2 Nios II IDE 集成开发环境 …… 235
6.1.3 GNU 工具链 …… 239
6.1.4 HAL 系统库 …… 240
6.1.5 RTOS 和 TCP/IP 协议栈 …… 240
6.2 HAL 系统库 …… 240
6.2.1 HAL 系统库结构 …… 240
6.2.2 使用 HAL 开发应用程序 …… 241
6.2.3 使用 HAL 开发设备驱动程序 …… 247
6.3 使用 Nios II IDE 建立用户应用程序 …… 251
6.3.1 创建一个新的 C/C++应用工程 …… 252
6.3.2 设置 C/C++应用工程系统属性 …… 256
6.3.3 编译链接工程 …… 258
6.3.4 调试程序 …… 260
6.3.5 运行程序 …… 261
6.3.6 使用 ModelSim 进行 RTL 级系统仿真 …… 263

**第 7 章 FPGA 配置和 Flash 编程**

7.1 FPGA 配置概述 …… 272
7.2 Cyclone II 系列 FPGA 配置 …… 276

7.2.1 Cyclone II 系列 FPGA 配置概述 …… 277
7.2.2 配置文件 …… 277
7.2.3 配置数据的压缩 …… 278
7.2.4 AS 配置模式 …… 279
7.2.5 PS 配置模式 …… 287
7.2.6 JTAG 配置模式 …… 298
7.2.7 Cyclone II 系列 FPGA 的配置引脚定义 …… 304
7.3 FPGA 的配置器件 …… 308
7.3.1 串行配置器件功能描述 …… 309
7.3.2 AS 模式下的 FPGA 配置 …… 312
7.3.3 AS 模式下串行配置器件的存储器访问 …… 313
7.3.4 电源与操作 …… 320
7.3.5 串行配置器件的时序 …… 321
7.3.6 编程与配置文件 …… 323
7.3.7 串行配置器件的引脚描述 …… 323
7.4 Quartus II 中 FPGA 配置的选项 …… 325
7.4.1 配置选项 …… 325
7.4.2 配置文件的格式 …… 329
7.5 FPGA 配置调试技术 …… 335
7.5.1 板子布局 …… 335
7.5.2 调试技术 …… 335
7.6 IDE Flash Programmer 介绍 …… 337
7.6.1 Nios II IDE Flash Programmer 的编程过程 …… 338
7.6.2 Flash 编程模式 …… 338
7.6.3 Flash Programmer 目标设计 …… 338
7.6.4 Flash 中的编程内容 …… 340
7.6.5 Flash 文件 …… 340
7.7 用户程序引导 …… 341
7.7.1 从 CFI Flash 中引导用户程序 …… 341
7.7.2 从 EPCS 串行配置器件中引导用户程序 …… 341
7.7.3 引导复制程序 …… 341
7.8 在 IDE 模式下使用 Nios II Flash Programmer …… 342
7.9 板子描述编辑器 …… 343
7.9.1 简 介 …… 343

7.9.2 板子描述的创建 …… 344
7.9.3 Flash Memory 标签 …… 346
7.9.4 Files 标签 …… 347

**第 8 章 Nios II 系统深入设计**

8.1 异常处理程序的开发 …… 349
8.1.1 Nios II 异常分类 …… 349
8.1.2 硬件抽象层的实现 …… 349
8.1.3 中断服务程序 …… 352
8.2 缓存和紧耦合存储器的编程 …… 357
8.2.1 复位后缓存的初始化 …… 358
8.2.2 设备驱动程序中缓存的编程 …… 359
8.2.3 装载程序或自更新程序的编写 …… 360
8.2.4 多主/多 CPU 系统中缓存的管理 …… 361
8.2.5 紧耦合存储器概述 …… 362
8.2.6 紧耦合存储器接口 …… 364
8.2.7 使用紧耦合存储器创建 Nios II 系统 …… 364
8.3 μC/OS-II 实时操作系统 …… 371
8.3.1 μC/OS-II 实时操作系统简介 …… 371
8.3.2 μC/OS-II 在 Nios II 上的移植 …… 372
8.3.3 Nios II IDE 中 μC/OS-II 工程的实现 …… 375
8.3.4 设计实例的软硬件要求 …… 380
8.3.5 在 Nios II 中建立 μC/OS-II 工程 …… 380
8.4 以太网与轻量 IP …… 385
8.4.1 轻量 IP 概述 …… 385
8.4.2 轻量 IP 协议栈的使用 …… 386
8.4.3 Nios II IDE 中轻量 IP 的配置 …… 391
8.4.4 设计实例的软硬件环境 …… 394
8.4.5 实例的设计文件 …… 394
8.4.6 软件开发流程 …… 395
8.4.7 Simple Socket Server 设计概述 …… 400
8.5 Nios II 多处理器系统 …… 408
8.5.1 多处理器系统设计概述 …… 408
8.5.2 多处理器系统的优势 …… 408

8.5.3 Nios II 多处理器系统硬件设计 …… 408
8.5.4 Nios II 多处理器系统的资源共享 …… 409
8.5.5 Nios II 多处理器系统软件设计 …… 413
8.5.6 Nios II 多处理器系统设计实例 …… 415
8.6 定制 Nios II 用户指令 …… 427
8.6.1 Nios II 定制指令综述 …… 427
8.6.2 定制指令实现方式 …… 433
8.6.3 定制指令设计实例 …… 436
8.7 定制基于 Avalon 的用户外设 …… 445
8.7.1 设计概述 …… 445
8.7.2 组件开发流程 …… 446
8.7.3 设计实例——PWM 从接口 …… 449

**第 9 章 基于 Cyclone II 开发板的 SOPC 系统开发实例**

9.1 全彩 LED 音乐景观灯控制系统总体设计方案 …… 460
9.1.1 系统需求分析 …… 460
9.1.2 硬件总体设计方案 …… 461
9.1.3 软件总体设计方案 …… 463
9.2 控制系统硬件设计与实现 …… 463
9.2.1 全彩 LED 灯控 PWM IP 核的设计与实现 …… 463
9.2.2 CF 卡设计与实现 …… 470
9.2.3 MP3 解码硬件电路设计与实现 …… 481
9.2.4 LED 发光二极管控制与驱动电路设计 …… 483
9.3 控制系统软件设计 …… 485
9.3.1 读取 CF 卡软件设计 …… 485
9.3.2 全彩 LED 灯控制程序设计 …… 488
9.3.3 播放 MP3 音乐软件设计 …… 491

**第 10 章 仿真与调试**

10.1 使用 Quartus II Simulator 进行仿真设计 …… 496
10.1.1 Quartus II Simulator 仿真 …… 496
10.1.2 Quartus II 支持的 EDA 仿真 …… 506
10.1.3 Quartus II 的延时分析 …… 509
10.2 使用 ModelSim 进行仿真 …… 512

10.2.1 ModelSim 仿真软件简介 …… 512
10.2.2 ModelSim 软件的结构 …… 514
10.2.3 ModelSim 运行方式 …… 514
10.2.4 ModelSim 的功能仿真 …… 515
10.2.5 门级仿真与时序仿真 …… 522
10.2.6 初始化存储器 …… 529
10.2.7 测试平台 …… 535
10.3 SignalTap II 实时测试 …… 542
10.3.1 SignalTap II 逻辑分析仪用户界面 …… 543
10.3.2 应用 SignalTap II 逻辑分析仪测试信号 …… 544
**附录 A Altera Cyclone II Nios II 实验开发套件** …… 548
**附录 B Cyclone II EP2C35 实验开发板** …… 550
**附录 C Cyclone II EP2C35 引脚表** …… 560
**参考文献** …… 564

# 第 1 章

# 概　述

本章简要介绍 SOPC 及其技术、Nios II 软核 SOPC 系统组件及开发环境。SOPC 及其技术包括嵌入式系统、SOC、FPGA/CPLD 等与 SOPC 技术密不可分的概念和技术。Nios II 软核 SOPC 系统组件包括 Nios II 软核嵌入式处理器、Avalon 总线和外围设备三部分。Nios II 软核 SOPC 系统开发环境包括硬件开发环境和软件开发环境。通过本章的学习，希望读者掌握 SOPC 的概念；了解 SOPC 的基本组成和总体结构；并对其软硬件开发环境和开发流程建立起整体概念。

## 1.1　SOPC 及其技术

本节对 SOPC 及与 SOPC 技术密切相关的嵌入式系统、SOC、FPGA/CPLD 等概念进行概述，简要说明 FPGA/CPLD 的基础知识和开发流程，目的是使读者对 SOPC 及其技术有一个较全面的了解和认识。为方便对本书所举实例的深入学习，本节的最后还对新一代低成本 FPGA——Cyclone II——的基本结构和性能进行了简要介绍。

### 1.1.1　嵌入式系统简介

嵌入式系统(Embedded System)无疑是当前最热门、最有发展前途的 IT 应用领域之一。嵌入式系统用在一些特定专用设备上，通常这些设备的硬件资源(如处理器、存储器等)非常有限，并且对成本很敏感，有时对实时响应要求很高等。特别是随着消费家电的智能化，嵌入式系统更显重要。像我们平常见到的手机、PDA、电子字典、可视电话、VCD/DVD/MP3 Player、数码相机(DC)、数码摄像机(DV)、U - Disk、机顶盒(Set Top Box)、高清电视(HDTV)、游戏机、智能玩具、交换机、路由器、数控设备或仪表、汽车电子、家电控制系统、医疗仪器、航天航空设备等都是典型的嵌入式系统。

所谓嵌入式系统，实际上是“嵌入式计算机系统”的简称，是相对于通用计算机系统而言的。根据IEEE(国际电气和电子工程师协会)的定义：嵌入式系统是用来控制或监视机器、装置或工厂等的大规模的设备。此定义是从应用方面考虑的，嵌入式系统是软件和硬件的综合体，是软件和硬件设计的完美结合。国内一般定义为：“以应用为中心，以计算机技术为基础，软硬件可裁剪，功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统”。广义上讲，凡是带有微处理器的专用软硬件系统都可称为嵌入式系统，如各类单片机和DSP系统。但由于它们没有操作系统，管理系统硬件和软件的能力有限，在实现复杂多任务功能时，往往困难重重，甚至无法实现。我们现在所说的嵌入式系统是指那些使用嵌入式微处理器构成独立系统，并将操作系统嵌入进去的嵌入式系统。

嵌入式系统的核心是嵌入式处理器，因此嵌入式处理器的技术指标如功耗、体积、成本、可靠性、速度、处理能力、电磁兼容性等均受到应用要求的制约，嵌入式处理器的应用软件是实现嵌入式系统功能的关键。一般地，软件要求固化存储，有时称为固件(Firmware)，软件代码要求高质量、高可靠性，大部分应用的系统软件(OS)的高实时性是基本要求。嵌入式系统与通用计算机系统形成了鲜明的对比，首先它们存在一些共同点，即其组成都包含了硬件和软件，但它们所完成的工作却截然不同。嵌入式系统往往只是一个大系统中的某个组成部分，控制大系统的工作，其价值在于它所控制的大系统，一般不取决于内嵌的处理器的性能指标。而通用计算机的功能和价值体现在“计算”上，计算能力、存储能力等是通用计算机的基本指标。不管什么型号的计算机，衡量它们的指标相同。随着嵌入式系统不断深入到人们生活的各个领域，嵌入式处理器得到了前所未有的飞速发展，目前据不完全统计，全世界嵌入式品种总量已超过1 500多种，流行体系结构有50多个系列。

虽然8位微控制器(主要是51单片机)仍然占据和继续领导标准低端嵌入式产品市场，以8位为核心的嵌入式技术不断发展，性能也在不断提高，但由于其性能的局限性，已无法满足高性能嵌入式技术的发展、提高。因此，以32位处理器作为高性能嵌入式系统开发的核心，已是嵌入式技术发展的必然趋势。

比较流行的高性能嵌入式RISC处理器主要有ARM公司的ARM系列、IBM公司的PowerPC、MIPS公司的MIPS、Sun公司的Sparc等。嵌入式操作系统主要有VxWorks、Windows CE、Linux、pSOS、EPOC、Palm OS、QNX、BeOS、VRTX、μC/OS、μCLinux等。

虽然有多种嵌入式处理器可供选择，但ARM处理器以其高性能、低功耗等突出优点已在32位嵌入式应用中稳居世界第一，已占据32位、64位嵌入式应用的绝大部分市场。ARM已成为移动通信、手持计算、多媒体数字消费等嵌入式解决方案事实上的标准。优良的性能和广泛的市场定位极大地增加和丰富了ARM的资源，加速了基于ARM处理器面向各种应用的系统芯片的开发和发展。

ARM系列处理器是英国先进RISC机器公司(Advanced RISC Machine，ARM)的产品。ARM公司是业界领先的知识产权供应商，只采用IP授权的方式允许半导体公司生产基于

ARM的处理器产品，提供基于ARM处理器内核的系统芯片解决方案和技术授权，不提供具体的芯片。目前非常流行的ARM内核有ARM7TDMI、StrongARM、ARM720T、ARM9TDMI、ARM920T、ARM940T、ARM946T、ARM966T、TARM10TDM等。ARM系列还获得了许多实时操作系统供应商的支持。

ARM的商业模式必将使得ARM获得更加广泛的应用，远远超出当初51发展的规模，确定了ARM技术和市场的领先地位，获得了更大的技术与商业成功。

互联网的普及、微电子加工工艺的提高、3C(Computer、Communication、Consumer)技术的普遍融合、信息服务应用的生活化等必将加速嵌入式技术的发展，以满足人们随时随地利用任何设备和手段来接收、处理和发布信息的需求，并将造就更加广阔的市场空间。图1-1所示是嵌入式技术发展的示意图，而SOC和SOPC代表了嵌入式系统发展的方向。

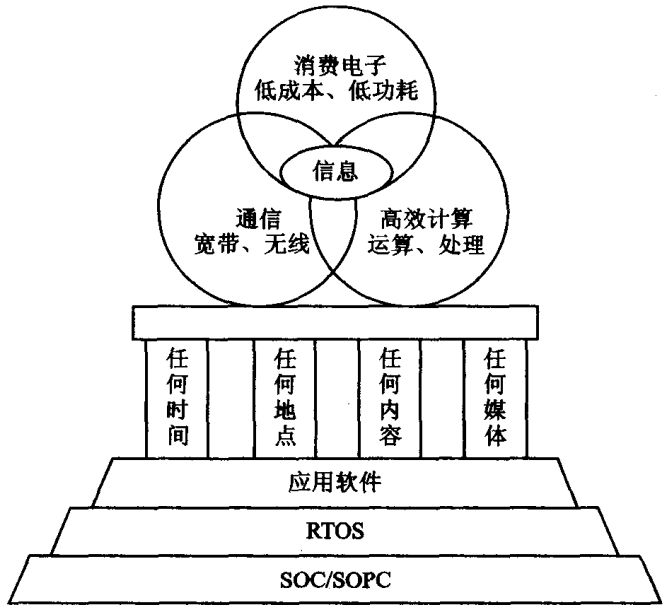


图1-1 嵌入式发展示意图

## 1.1.2 SOC系统简介

20世纪90年代后期，嵌入式系统设计从以嵌入式微处理器/DSP为核心的“集成电路”级设计不断地转向“集成系统”级设计，提出了SOC(System on a Chip，片上系统)的基本概念。目前嵌入式系统已进入了单片系统SOC的设计阶段，并逐步进入实用化、规范化阶段，集成电路已进入SOC的设计流程。集成电路和系统功能达到什么程度才算SOC，并没有严格的定义，但广义而言，SOC应该指在单片上集成系统级多元化的大规模功能模块，从而构成一个能