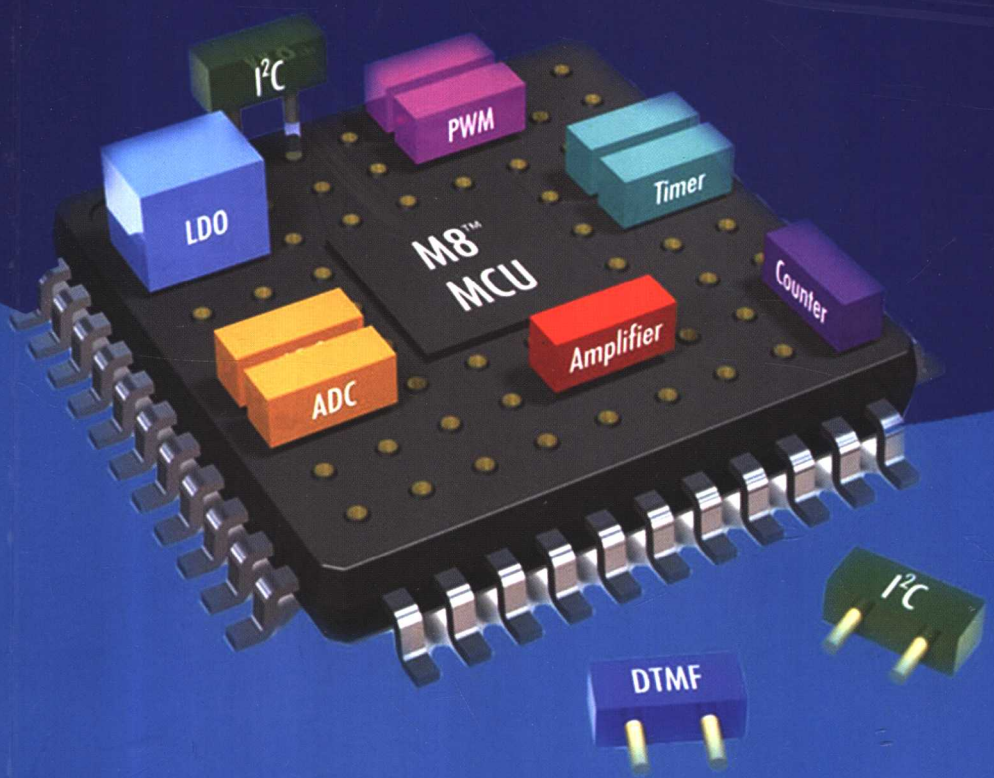
CYPRESS
PERFORM

PSoC™ Architecture and Programming

PSoC™ 体系结构与编程

戴国骏　张 翔　曾 虹



中国科学技术出版社

PsoC™体系结构与编程

戴国骏　张 翔　曾 虹

中国科学技术出版社
·北　京·

图书在版编目(CIP)数据

PSOC™体系结构与编程/戴国骏,张翔,曾虹.北京:中国科学技术出版社,2005.9

ISBN 7-5046-4172-3

Ⅰ.P... Ⅱ.①戴... ②张... ③曾... Ⅲ.微处理器,PSOC-系统结构-程序设计 Ⅳ.TP332

中国版本图书馆 CIP 数据核字(2005)第 110437 号

中国科学技术出版社出版
北京市海淀区中关村南大街 16 号 邮政编码:100081
电话:010-62103210 传真:010-62183872
科学普及出版社发行部发行
北京玥实印刷有限公司印刷

*

开本:787 毫米×1092 毫米 1/16 印张:20.75 字数:250 千字
2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷
印数:1-5000 册 定价:36.00 元

内 容 简 介

PSoC™ 系列芯片作为一种新型的片上可编程系统芯片（SoPC），片内集成了可编程的数字与模拟系统，可灵活配置用户所需的各种功能模块，是第一种真正具有混合信号处理能力的片上系统（SoC）。本书详细阐述了 PSoC™ 芯片的硬件体系结构，最有特色的可编程数字系统和模拟系统，指令系统与汇编语言，C 语言编程，具有软硬件协同设计功能的集成开发环境，分析了目前 PSoC™ 的 CY8C21/24/27/29 系列芯片的结构特点、系统资源与封装，并给出了相应的典型应用实例。本书力求在体系结构上论述清晰，在器件应用上具体而详实。

本书主要面向嵌入式计算机系统研究开发的工程技术人员，可作为计算机、电子工程、工业自动化等领域工程技术人员的设计参考书，也可作为高等学校相关专业教材或教学参考书。

前　言

随着大规模集成电路技术的不断发展，嵌入式计算机系统开始从MCU逐步过渡到SoC的新阶段。PSoC™ 是由美国赛普拉斯半导体公司（Cypress Semiconductor, 以下简称Cypress）首推完全基于通用IP模块，具有真正混合信号处理能力的可编程片上系统芯片。我们相信当你真正了解和掌握了 PSoC™ 并应用到具体的嵌入式系统中，将永远不会再使用那些过时、枯燥的 MCU 进行设计工作。本书的目的就是向读者展示 PSoC™ 的各种独特之处，了解 SoC 的发展趋势。

本书首先从系统结构的角度去把握 PSoC™ 芯片，详细阐述了其 M8C 内核，最具有特色的可编程数字和模拟系统及相应的可编程互联总线，PSoC™ 丰富的系统资源；从系统程序员角度介绍各种可配置数字、模拟及混合信号处理功能模块的 API 函数，同时介绍了 PSoC™ 丰富的开发手段，如汇编器、C 编译器及集成开发环境，最后给出几个实际的例子，希望通过本书把读者引入到 PSoC™ 这一代表嵌入式系统发展方向充满创新气息的 SoC 的开发应用。

本书内容安排如下：第一章为 PSoC™ 概述，第二章为 PSoC™ 体系结构，第三章为 PSoC™ 可编程数字系统，第四章为 PSoC™ 可编程模拟系统，第五章为 PSoC™ 汇编语言编程，第六章为 PSoC™ C 语言编程，第七章为 PSoC™ 集成开发环境，第八章为 PsoC ™ CY8C21 系列芯片及应用，第九章为 PSoC™ CY8C24 系列芯片及应用，第十章为 PSoC™ CY8C27/29 系列芯片及应用，第十一章为不断创新的 PSoC™。

本书主要由戴国骏、张翔、曾虹负责编写；计算机应用研究所的戴渊明、方江江、高申勇、葛海江、李二涛、刘世杰、汤孙寿、姚利华参与了具体工作并贡献了他们的研究成果。在编写的过程中得到 Cypress 上海办事处销售经理陈文隆先生和应用工程师刘辉先生及香港德创电子有限公司上海代表处经理黄宁先生的大力协助。浙江大学电气工程学院沈建新教授对本书提出了宝贵的修改意见，在此表示衷心感谢。为了使本书更具有全面性、实用性，在编写的过程中我们查阅了大量的资料，因篇幅有限，难以一一列举，在此向各原作者表示感谢。特别感谢杭州电子科技大学学报主任傅恒及杭州理想电子有限公司何秀育、袁孝炯为本书所做的大量具体工作。

本书的编写本着全面性、实用性的原则，以详尽的实例帮助读者以最短的时间，准确、有效地融入 PSoC™(正文统一简称为 PSoC)的开发潮流中；尽管编者尽了最大的努力，但因时间仓促，加之编者的水平有限，不妥之处在所难免，恳请读者批评指正。需要本书源程序的读者请与作者联系。

戴国骏

2005 年 6 月 30 日

于杭州电子科技大学

Preface

PSoC is a modern design platform for system designers.

Traditional choices for system designer are:

- Fixed function product
- Microcontroller
- ASIC

When you choose fixed function part, you get some simplicity, but very little flexibility. When you choose a microcontroller based design, you get flexibility, but longer design cycle. When you choose an ASIC, you get best cost, but longer design cycle and an NRE and no flexibility. PSoC is a bridge that gives you the best of each. PSoC is a configurable mixed signal array that provides you a micro-controller and configurable analog and digital blocks.

A single PSoC device can integrate as many as 100 peripheral functions with a microcontroller, saving customers design time, board space, power consumption and from 5 cents to as much as $10 in system costs. Easy-to-use development tools enable designers to select the precise peripheral functionality they desire, including analog functions such as amplifiers, ADCs, DACs, filters and comparators and digital functions such as timers, counters, PWMs, SPI and UARTs.

The PSoC family's analog features include rail-to-rail inputs, programmable gain amplifiers and up to 14-bit ADCs with exceptionally low noise, input leakage and voltage offset. PSoC devices include up to 32 KB of flash memory, 2 KB of SRAM, an 8×8 multiplier with 32-bit accumulator, power and sleep monitoring circuits, and hardware I^2C communications.

PSoC Designer(TM), the traditional software development environment for PSoC, is a full-featured, GUI-based design tool suite that enables the user to configure design-in silicon with simple point and click options. With PSoC Designer, users can code the MCU in either 'C' or Assembly language and debug the design using sophisticated features such as event triggers and multiple break points, while single stepping through code in 'C' or Assembly or a mix of the two.

Newly introduced, PSoC Express is the first development tool that allows system engineers to develop microcontroller-based designs at a level that does not require any Assembly language or 'C' programming." By operating at a higher level of abstraction than previously possible and removing the necessity to develop the required firmware, PSoC Express enables new designs to be created, simulated and programmed to the targeted PSoC device in hours or days instead of in weeks or months. With PSoC Express, designers work within their areas of application expertise, defining a custom solution by choosing input and output devices from a catalog, and then logically linking them to define system behavior. Within PSoC Express the designer is able to verify designs through simulation, then generate and download the device-programming file. The new tool also creates customized project documentation including a data sheet with register map, interface schematics, and bill of materials. Without writing any microcontroller code, designers implement reliable, custom applications faster than they can write the requirements.

PSoC comes in price points of $0.50 to $3.00 and you can move projects from one device to another with simple recompile.

We hope you find a simple, flexible, inexpensive solution to your design project with PSoC.

George Saul
CEO Subsidiary Cypress(CMS)
Sep. 8, 2005

目　录

前　言

第一章　PSoC 概述 ····· 1

1.1　**系统集成芯片 SoC** ····· 1

1.1.1　SoC 的概述 ····· 1

1.1.2　SoC 的设计 ····· 1

1.1.3　SoC 中 IP 核的设计和复用 ····· 3

1.1.4　SoC 设计方法所面临的挑战 ····· 4

1.1.5　SoC 技术发展趋势 ····· 5

1.2　**系统集成可编程芯片(SoPC)** ····· 6

1.2.1　Actel 公司的 VariCore 核 ····· 7

1.2.2　Altera 公司的 Excalibur 核 ····· 8

1.2.3　Atmel 公司的 AVR8 核 ····· 8

1.2.4　QuickLogic 公司的 MIPS 核 ····· 9

1.2.5　Xilinx 公司的 PowerPC 核 ····· 10

1.2.6　Triscend 公司的 8051/ARM7 核 ····· 10

1.3　**PSoC 的结构及特点** ····· 11

1.3.1　模拟和数字结合的可配置 SoC ····· 11

1.3.2　PSoC 的总体结构 ····· 12

1.3.3　PSoC 可编程数字系统和模拟系统 ····· 14

1.3.4　PSoC 的系统资源 ····· 15

1.3.5　PSoC 的开发工具 ····· 15

1.4　**PSoC 和单片机系统** ····· 16

1.4.1　单片机的发展历程 ····· 16

1.4.2　PSoC 与传统单片机系统设计方案的比较 ····· 17

1.5　**PSoC 设计开发流程** ····· 18

1.6　**PSoC 系列的特点及选择** ····· 19

第二章　PSoC 体系结构 ····· 21

2.1　**PSoC 的总体结构** ····· 21

2.1.1　PSoC 内核 ····· 22

2.1.2　数字系统 ····· 22

2.1.3 模拟系统 · · · · · 23
2.1.4 系统资源 · · · · · 23
2.2 PSoC 内核 · · · · · 24
2.2.1 M8C · · · · · 24
2.2.2 监控 ROM · · · · · 25
2.2.3 RAM 分页 · · · · · 32
2.2.4 中断控制器 · · · · · 36
2.2.5 通用输入输出 · · · · · 39
2.2.6 模拟输出驱动 · · · · · 44
2.2.7 内部主振荡器 · · · · · 44
2.2.8 内部低速振荡器 · · · · · 45
2.2.9 外部晶体振荡器 · · · · · 46
2.2.10 锁相环 · · · · · 47
2.2.11 睡眠和看门狗 · · · · · 47
2.3 系统资源 · · · · · 49
2.3.1 数字时钟 · · · · · 50
2.3.2 乘法加法器 · · · · · 54
2.3.3 采样抽取器 · · · · · 56
2.3.4 I^2C 通信模块 · · · · · 58
2.3.5 内部参考电压 · · · · · 60
2.3.6 系统复位 · · · · · 61
2.3.7 开关式升压泵 · · · · · 61
2.3.8 上电复位和低电压检测 · · · · · 62
2.3.9 I/O 模拟多路复用器 · · · · · 63
2.3.10 全速的 USB · · · · · 64
2.4 寄存器分类概述 · · · · · 68
2.4.1 PSoC 内核寄存器分类概述 · · · · · 69
2.4.2 PSoC 数字系统寄存器分类概述 · · · · · 69
2.4.3 PSoC 模拟系统寄存器分类概述 · · · · · 69
2.4.4 PSoC 系统资源寄存器分类概述 · · · · · 69

第三章 PSoC 可编程数字系统 · · · · · 70
3.1 PSoC 数字系统体系结构 · · · · · 70
3.1.1 全局数字系统互连(GDI) · · · · · 71
3.1.2 行间数字阵列互连(ADI) · · · · · 72
3.1.3 行内数字模块互连(RDI) · · · · · 74
3.1.4 数字 PSoC 基本模块内部体系结构 · · · · · 74

3.2 定时器和计数器功能模块 · · · · · 75
3.2.1 结构和功能 · · · · · 75
3.2.2 参数配置和 API 函数 · · · · · 77
3.2.3 应用举例 · · · · · 81
3.3 数字脉宽调制(PWM)模块 · · · · · 82
3.3.1 结构和功能 · · · · · 82
3.3.2 参数配置和 API 函数 · · · · · 84
3.3.3 应用举例 · · · · · 86
3.4 串行通信端口 SPI · · · · · 87
3.4.1 结构和功能 · · · · · 87
3.4.2 参数配置和 API 函数 · · · · · 89
3.4.3 应用举例 · · · · · 94
3.5 串行通信端口 UART · · · · · 94
3.5.1 结构和功能 · · · · · 94
3.5.2 参数配置和 API 函数 · · · · · 97
3.5.3 应用举例 · · · · · 107
3.6 其他数字模块 · · · · · 109
3.6.1 EEPROM 模块 · · · · · 109
3.6.2 DigBuf 模块 · · · · · 114

第四章 PSoC 可编程模拟系统 · · · · · 116
4.1 PSoC 模拟系统体系结构 · · · · · 116
4.1.1 PSoC 模拟系统总体体系结构 · · · · · 116
4.1.2 全局模拟互连(GAI) · · · · · 117
4.1.3 基本模拟 PSoC 模块阵列 · · · · · 118
4.1.4 模拟系统输入信号选择器 · · · · · 118
4.1.5 模拟信号基准电压发生器 · · · · · 123
4.1.6 模拟 PSoC 模块内部体系结构 · · · · · 123
4.2 模数转换器功能模块 · · · · · 125
4.2.1 6 位模数转换器 SAR6 功能模块 · · · · · 126
4.2.2 12 位模数转换器 ADCINC12 功能模块 · · · · · 128
4.2.3 8 位模数转换器 DELSIG8 功能模块 · · · · · 133
4.2.4 8 位模数转换器 ADC8 功能模块 · · · · · 138
4.3 数模转换器功能模块 · · · · · 143
4.3.1 8 位数模转换器 DAC8 功能模块 · · · · · 143
4.4 放大器功能模块 · · · · · 147
4.4.1 增益可编程放大器 PGA 功能模块 · · · · · 147

4.4.2 基准电压可编程比较器 CMPPRG 功能模块 150
4.5 滤波器功能模块 153
4.5.1 双极点带通滤波器 BPF2 功能模块 153
4.5.2 双极点低通滤波器 LPF2 功能模块 156

第五章 PSoC 汇编语言编程 160
5.1 M8C 内核处理器 160
5.1.1 M8C 简介 160
5.1.2 内部寄存器 160
5.1.3 地址空间 161
5.1.4 指令格式 162
5.1.5 寻址模式 163
5.2 M8C 指令集 167
5.2.1 算术运算类指令 167
5.2.2 逻辑运算类指令 171
5.2.3 移位类指令 173
5.2.4 数据传送类指令 178
5.2.5 转移控制类指令 182
5.2.6 处理器类指令 188
5.3 PSoC 汇编语言和汇编编译器 191
5.3.1 源文件 191
5.3.2 目录文件 195
5.3.3 图文件 195
5.3.4 ROM 文件 196
5.3.5 Intel HEX 文件 196
5.3.6 内部寄存器的恢复 198
5.4 编译器伪指令系统 198
5.5 PSoC 汇编程序实例 205
5.5.1 创建工程 206
5.5.2 文件分类 209
5.5.3 boot 和 main 源文件 209

第六章 PSoC C 语言编程 215
6.1 PSoC C 语言的数据类型与操作符 215
6.1.1 数据类型 215
6.1.2 操作符 216
6.1.3 位操作 217

6.2 PSoC C 语言的控制语句 · · · · · · 218
6.2.1 条件语句 · · · · · · 218
6.2.2 循环语句 · · · · · · 219
6.2.3 switch 语句 · · · · · · 220
6.2.4 break、continue 和 goto 语句 · · · · · · 221
6.2.5 return 语句 · · · · · · 222
6.3 PSoC C 语言指针 · · · · · · 222
6.3.1 指针的概念 · · · · · · 222
6.3.2 指针变量的赋值 · · · · · · 222
6.3.3 指针变量的运算 · · · · · · 223
6.4 PSoC C 编译器及库函数 · · · · · · 224
6.4.1 PSoC C 语言编译器 · · · · · · 224
6.4.2 预处理语句 · · · · · · 224
6.4.3 PSoC C 库函数 · · · · · · 225
6.5 PSoC C 语言编程示例 · · · · · · 225
6.5.1 创建工程 · · · · · · 225
6.5.2 编写 PSoC C 语言程序 · · · · · · 228

第七章 PSoC 集成开发环境 · · · · · · 230
7.1 PSoC Designer 的安装 · · · · · · 230
7.1.1 软硬件要求 · · · · · · 230
7.1.2 安装步骤 · · · · · · 231
7.1.3 激活 PSoC C 语言编译器 · · · · · · 235
7.2 PSoC IDE 的使用 · · · · · · 235
7.2.1 PSoC IDE 的结构 · · · · · · 235
7.2.2 文件类型和扩展名 · · · · · · 236
7.2.3 工程管理器 · · · · · · 237
7.2.4 创建工程 · · · · · · 238
7.2.5 创建工程的方法 · · · · · · 240
7.3 器件编辑器 · · · · · · 241
7.3.1 选择用户模块 · · · · · · 242
7.3.2 放置用户模块 · · · · · · 243
7.3.3 配置用户模块 · · · · · · 243
7.3.4 其他 · · · · · · 244
7.4 连接用户模块 · · · · · · 246
7.5 管脚互连 · · · · · · 251
7.5.1 连接管脚 · · · · · · 253

7.5.2 端口驱动模式 · · · · · 255
7.5.3 端口中断模式 · · · · · 255
7.6 应用程序编辑器 · · · · · 256
7.7 调试 · · · · · 258
7.7.1 调试组件 · · · · · 258
7.7.2 连接软硬件 · · · · · 258
7.7.3 下载到 Pod · · · · · 259
7.7.4 调试策略 · · · · · 259
7.7.5 调试工具栏和图标 · · · · · 261
7.7.6 烧写芯片 · · · · · 261

第八章 PSoC CY8C21 系列芯片及应用 · · · · · 263
8.1 PSoC CY8C21 系列芯片简介 · · · · · 263
8.1.1 系统资源 · · · · · 263
8.1.2 封装 · · · · · 263
8.2 非接触电容开关 · · · · · 264
8.2.1 电容开关原理 · · · · · 264
8.2.2 电容值测量方法 · · · · · 264
8.3 PSoC 的非接触电容开关面板 · · · · · 267
8.3.1 面板说明 · · · · · 267
8.3.2 PSoC 方案简介 · · · · · 267

第九章 PSoC CY8C24 系列芯片及应用 · · · · · 273
9.1 PSoC CY8C24 系列芯片简介 · · · · · 273
9.1.1 系统资源 · · · · · 273
9.1.2 封装 · · · · · 274
9.2 直流电动机控制原理 · · · · · 274
9.2.1 直流电机原理和模型 · · · · · 274
9.2.2 直流电动机的驱动电路 · · · · · 276
9.3 PSoC 直流电动机控制系统 · · · · · 277
9.3.1 系统设计 · · · · · 278
9.3.2 PSoC 系统实现 · · · · · 279

第十章 PSoC CY8C27/29 系列芯片及应用 · · · · · 281
10.1 CY8C27/29 系列芯片简介 · · · · · 281
10.1.1 CY8C27 系列芯片简介 · · · · · 281
10.1.2 CY8C29 系列芯片 · · · · · 282
10.2 钻尾机的控制 · · · · · 283

10.2.1 工作模式检测 · · · · · · 283
10.2.2 油路系统检测 · · · · · · 283
10.2.3 信号系统检测 · · · · · · 284
10.2.4 液晶显示 · · · · · · 284
10.3 PSoC 的钻尾机控制系统 · · · · · · 285
10.3.1 系统的硬件设计方案 · · · · · · 286
10.3.2 系统的软件设计方案 · · · · · · 287
10.3.3 CY8C27443 内部资源配置 · · · · · · 289
第十一章 不断创新的 PSoC · · · · · · 293
11.1 新一代开发环境 PSoC Express · · · · · · 293
11.2 带全速 USB 的 PSoC · · · · · · 295
11.3 带 Wireless USB 的 PRoC · · · · · · 296
11.3.1 近距离无线通讯技术 · · · · · · 296
11.3.2 Wireless USB 无线技术 · · · · · · 297
11.3.3 带 Wireless USB 的 PRoC · · · · · · 298
11.4 宽工作电压的 PSoC · · · · · · 299
小 结 · · · · · · 300
附录 功能寄存器一览表 · · · · · · 301
参考文献 · · · · · · 315

第一章 PSoC 概述

1.1 系统集成芯片 SoC

1.1.1 SoC 的概述

将整个电子系统集成在同一芯片上，称为片上系统(System on Chip，简称 SoC)，或称为系统级芯片。SoC 技术以超深亚微米 UDSM(Ultra-Deep Submicron)工艺、知识产权核 IP Core(Intellectual Property Core)复用和软硬件协同设计技术为支撑，是当今超大规模集成电路 VLSI 的发展趋势，对微电子技术及其应用领域是一种革命性的变革。SoC 把处理器核、存储器、高密度逻辑电路、模拟和混合信号电路、DSP 预处理及其他功能的电路集成到一个芯片上，在芯片内部实现信号采集、转换、存储、处理和 I/O 接口以及电子产品所需要的所有的功能，形成片上系统。从应用的角度划分 SoC 有 3 种类型:专用集成电路 ASIC(Application Specific IC)型 SoC、可编程 SoC(System on Programmable Chip，SoPC)和 OEM(Original Equipment Manufacturer，原始设备生产商)型 SoC。21 世纪集成电路技术应用的主流将是可提供更好的性能、更低的功耗、更小的印制板空间和更低的成本的 SoC，SoC 技术的研究发展和应用对社会信息化建设有重大意义。

1.1.2 SoC 的设计

1. 集成系统(IS)与集成电路的比较

SoC 的设计思想与传统的集成电路的设计思想不同。SoC 的设计称之为集成系统的设计，集成系统和集成电路的关系相当于集成电路和分立元件的关系。在集成系统设计中，设计者面对的不再是电路芯片，而是能实现设计功能的 IP 核库。设计者不必要在众多的模块电路中搜索所需要的电路芯片，只需要根据设计功能和固件特性，选择相应的 IP 核。那些芯片面积小、运行速度最快、功率消耗最低、工艺容差最大的 IP 核具有很大的 IP 价值，这些 IP 核，将会被集成系统复用。集成系统设计的这种电路设计技术和综合方法，基本上消除了器件信息障碍，利用已有的 IP 核进行设计复用，这种设计方法从传统的集成电路设计转向集成系统设计，从整个系统的角度出发，把处理机制、模型算法、芯片结构、各层次电路直至器件的设计紧密地结合在一起，在单个芯片上完成整个系统的功能，设计的重心也从逻辑综合、布局布线转向系统的设计、软硬结合的设计以及仿真，它的设计必须是从系统行为级开始的自上向下的设计方法。

由于集成系统设计的方法能够综合并全盘考虑整个系统的各种情况，可以在同样的工艺技术条件下实现更高性能的系统指标。与集成电路设计方法相比，采用集成系统的设计方法完成同样功能所需的晶体管数目可以降低 2～3 个数量级。集成系统设计方法依靠的是广泛的知识背景。软、硬 IP 核设计中更多体现电路、器件、物理、工艺，甚至分子、原子等物理背景，而集成系统设计将更多地体现功能、行为、算法、架构、甚至思路、构想等系统背景。集成电路设计方法向集成系统设计方法的转变，不仅是一种概念上的突破，同时也是信息技术发展的必然结果，它对电子技术的推动将不亚于 20 世纪 50 年代集成电路设计技术。

2. SoC 的设计过程

用 SoC 技术设计应用电子系统的几个阶段如图 1.1 所示。在功能设计阶段，设计者必须充分考虑系统的固件特性，并利用固件特性进行综合功能设计。当功能设计完成后，就可以进入 IP 综合阶段。IP 综合阶段的任务是利用强大的 IP 核库实现系统的功能。IP 综合结束后，进行功能仿真，以检查是否实现了系统的设计功能要求。功能仿真通过后，就是电路仿真，目的是检查 IP 核组成的电路能否实现设计功能并达到相应的设计技术指标。设计的最后阶段对制造好的 SoC 产品进行相应的测试，以便调整各种技术参数，确定应用参数。

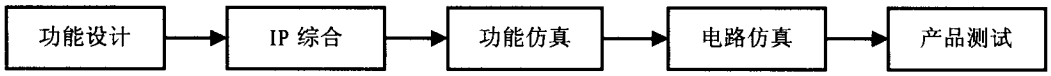


图 1.1　SoC 的设计过程

3. SoC 的设计方法

片上系统设计的 3 种不同方法：

(1) 专用片上系统设计方法(Costumes Design)。如果采用这种方法，系统厂商只需定义出系统指标，芯片中采用的处理器核心(MCU Core)、专用数字信号处理器核心(DSP Core)和专用电路部分全由芯片厂商完成，也可能由专门设计芯片的公司完成。这种方法得到的系统芯片成本可能最低，但需要较长的设计时间，系统厂商的灵活性较小，只适合用于片上系统产量特别大的情况。

(2) 部分集成法。系统厂商设计片上系统中的专用电路 MCU 核心、DSP 核心以及存储器等。IP 核由专用芯片厂商提供，并由芯片厂商负责把系统厂商设计的专用电路与各种 IP 核集成到片上系统中。这种方法有一定的灵活性，新产品开发时间较短。

(3) 桌面集成法。按照这种方法，各种 IP 核的设计者将 IP 核提供给电子系统厂商，由电子系统厂商设计其专用电路部分，并将专用电路与各种 IP 核集成在一起后交给芯片制造厂商制造出片上系统。这种设计方法与普通的半定制方法类似，设计成本最低，设计灵活性最大。

大部分片上系统仍采用第 1 种或者第 2 种方法，将各种核心模块由系统厂商在桌面集成还需要解决一些技术问题。第一，IP 核的开发者需要保护其技术利益，比如核心单元的

网表或者版图，这样就不太容易为系统厂商提供精确的模型。实际上，为了使各种 IP 核被系统厂商所采用，必须在不公开 IP 核网表的前提下，构造出 IP 核足够精确的仿真模型。第二，对于系统厂商来讲，通常需要对 IP 核的组成部分做一些取舍，以满足特定系统的要求。比如对一个特定的系统来讲，DSP 核心中的某些运算单元可能并不需要，应该在 IP 核开发者提供的 IP 核中去掉这些单元，以优化片上系统的面积和功耗。为了满足这些要求，IP 核开发者必须在不暴露 IP 核的核心细节的前提下，为系统厂商提供修改 IP 核的方法。第三，如果采用可综合的软核，片上系统设计者仍需对这些软核进行综合、测试、时序分析以及其他验证。功能和时序的再验证仍是复杂的工作，并且要对 IP 核的行为有较深入的了解。事实上，系统厂商对 IP 核的详细行为并不能深入地了解。第四，在微电子领域，面向市场的开发人员，很难事先搞清楚一个 IP 核能在多大程度上引起客户的兴趣，以及采用了这个 IP 核设计出的产品，最终有多大的用户需求。最后，在获得进行产品开发所需的核心技术之前，IP 核的设计者还要支付不小的前期费用。上面的这些因素，使开发者举步维艰，在一定程度上，阻碍了 IP 核的复用，限制了系统级芯片的开发。

1.1.3 SoC 中 IP 核的设计和复用

为什么要采用可复用 IP 核的设计方法来设计片上系统，做一个简单的计算，就可以知道为什么是这样。如果不用可复用的设计方法，一位好的设计师每天可以设计 100 个逻辑门或者 30 行 RTL 代码，这个数据在过去 5 年中一直是这样的，将来可能会发生变化。对于一个 100k 门的 ASIC 设计(20 世纪 90 年代的典型规模)，意味着需要 1 000 个设计日/人或 5 个人 200 天的工作量。对于一个 10M 门的设计，需要 10 万个工作日/人或 500 人 200 天的工作量，对于任何一块芯片来说，由此产生的费用都是不可承受的。如果采用偶然的可复用模块，效率只能提高 2~3 倍。因此，只有选择可复用的 IP 核来设计 SoC，才能真正解决 SoC 芯片设计所面临的问题。一个好的 IP 核，应该从两个方面考虑：(1)可用性设计；(2) 可复用设计。

SoC 系统设计的成败，直接依赖于 IP 核的质量和将核集成的工具。因为 IP 核是预先设计好的，并经过了验证，设计者可以把注意力集中于整个系统，而不必考虑各个核的正确性。使用 IP 核可以简化系统设计，缩短设计时间。

IP 核可分为软核(Soft Core)、硬核(Hard Core)、固核(Firm Core) 3 种。软核是用硬件描述语言(Verilog HDL 和 VHDL)的形式描述功能块的行为，并不涉及用什么电路和电路元件实现这些行为。软核可经用户修改，以实现所需的功能。由于软核没有经过具体电路的验证，使其在应用到具体电路的时候，可能有不符合要求的方面。硬核是在软核的基础上，结合半导体工艺、设计规则而生成的集成电路版图和网表。通常把软核写入芯片中，并通过调试、仿真通过的设计文件、程序、网表、版图的集合称为硬核。它的电路布局布线和工艺是确定的且不能更改，其经过投片验证，正确性较好。固核是一种介于软核和硬核之