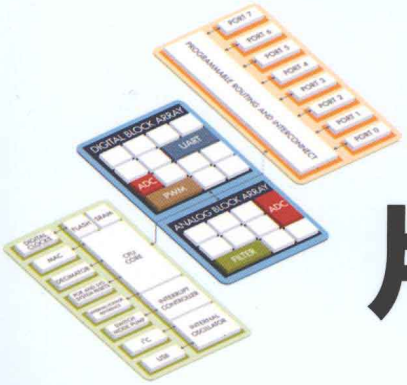
UNIVERSITY ALLIANCE PROGRAM

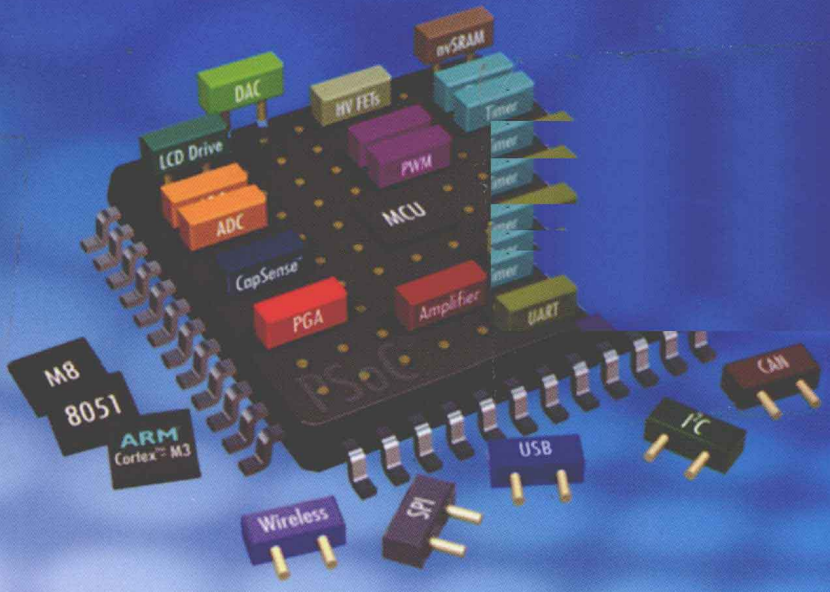
Cypress PSoC设计指南系列



8051 片上可编程系统 原理及应用

The principle and Application of 8051 Programmable System-on-Chip

▸▸▸ 何 宾 编著



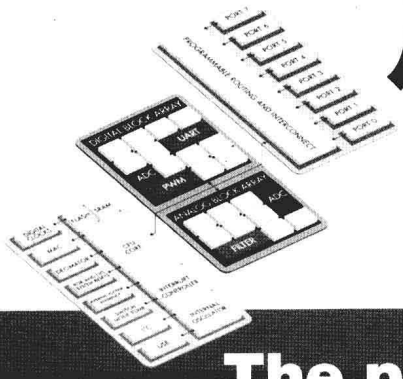
CD-ROM
配光盘
(源程序+课件)

化学工业出版社

THE CYPRESS UNIVERSITY ALLIANCE PROGRAM

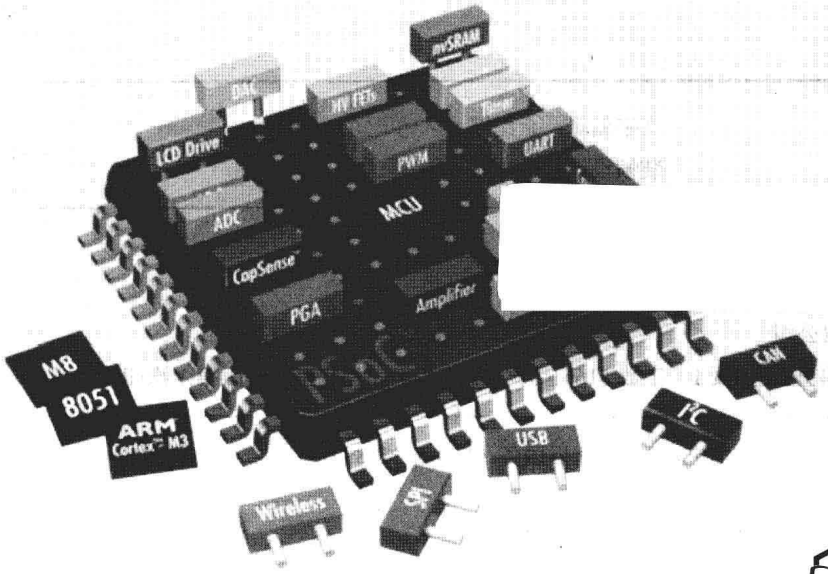
Cypress PSoC设计指南系列

8051片上可编程系统原理及应用



The principle and Application of 8051 Programmable System-on-Chip

▸▸▸ 何 宾 编著



CD-ROM
配光盘
(源程序+课件)

化学工业出版社
·北 京·

图书在版编目（CIP）数据

8051 片上可编程系统原理及应用 / 何宾编著. —北京：化学工业出版社，2012.2

ISBN 978-7-122-12675-7

Ⅰ. 8… Ⅱ. 何… Ⅲ. 单片微型计算机 Ⅳ. TP368.1

中国版本图书馆 CIP 数据核字（2011）第 217320 号

责任编辑：宋 辉　　文字编辑：余纪军

责任校对：王素芹　　装帧设计：韩 飞

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号 邮政编码 100011）

印　　刷：北京永鑫印刷有限责任公司

装　　订：三河市万龙印装有限公司

787mm×1092mm 1/16 印张 19½ 插页 1 字数 516 千字 2012 年 3 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686） 售后服务：010-64518899

网　　址：http: // www. cip. com. cn

凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：59.00 元

前　言

随着半导体技术的发展和芯片集成度的提高，越来越多的厂商开始提供在单芯片上实现复杂系统的解决方案，即基于 PSoC 的解决方案。这种解决方案提高了设计的可靠性，缩短了系统设计周期，降低了设计成本，极大地满足了市场对产品竞争力的要求。

作为全球知名的半导体公司——美国 Cypress 公司，率先在业界实现了完全意义上的 PSoC 解决方案，即在单芯片上实现了 MCU、数字和模拟系统的高度集成。PSoC 技术的不断发展，将大大推动电子系统设计方法的创新，并且对未来嵌入式系统设计领域带来深远的影响。Cypess 的 PSoC3 集成了 8051 CPU 核。这种集成 8051 CPU 的片上可编程系统，大大拓宽了 MCU 的应用领域，使得 MCU 焕发出新的生命力。

本书全面系统地介绍了 Cypress 公司的 PSoC3 可编程片上系统体系结构。通过介绍其系统结构和设计流程，使读者能全面地掌握 PSoC 的体系结构和实现方法。本书主要包括以下 17 部分。

（1）PSoC 设计导论部分。该部分内容包括微控制器基础、可编程片上系统 PSoC 概述、PSoC3 设计流程、PSoC3 的结构及功能、PSoC3 器件概述等内容。

（2）PSoC3 CPU 子系统部分。该部分内容包括 PSoC3 CPU 内核功能单元、PSoC3 存储器结构和地址空间、DMA 和 PHUB 结构及功能、中断控制器结构及功能等内容。

（3）PSoC3 CPU 指令系统部分。该部分内容包括 PSoC3 CPU 寻址模式、PSoC3 CPU 指令集、汇编语言编程模型等内容。

（4）PSoC3 公共资源部分。该部分内容包括时钟管理、电源管理、复位、I/O 系统和布线资源等内容。

（5）PSoC 编程和调试接口功能部分。该部分内容包括测试控制器、8051 片上调试、非易失性存储器编程等内容。

（6）基于 PSoC Creator 的程序设计部分。该部分内容包括 PSoC Creator 软件功能、GPIO 控制程序的设计、中断服务程序的设计等内容。

（7）定时器、计数器和 PWM 模块部分。该部分内容包括定时器模块、计数器模块、PWM 模块、PWM 控制 LED 显示的实现等内容。

（8）LCD 显示驱动模块部分。该部分内容包括 LCD 的工作原理、LCD 驱动接口概述、LCD 操作、段式 LCD 显示的实现等内容。

（9）I^2C 总线模块部分。该部分内容包括 I^2C 总线模块概述、I^2C 总线实现原理、I^2C 总线寄存器及操作、I^2C 总线操作模式、I^2C 模块通信的实现等内容。

（10）CAN 总线模块部分。该部分内容包括 CAN 总线模块概述、CAN 消息帧类型及格式、CAN 总线消息发送、CAN 总线消息接收、远程帧传输、位时间配置、错误处理及中断、CAN 总线通信的实现等内容。

（11）USB 总线模块部分。该部分内容包括 USB 总线模块概述、USB 模块结构、USB 模块

工作条件、逻辑传输模式、PS/2 和 CMOS I/O 模式、USB 人体学输入设备的实现等内容。

（12）通用数字块 UDB 部分。该部分内容包括通用数字块概述、PLD 模块、数据通道模块、状态和控制模块、基于 PLD 的自定义元件设计等内容。

（13）模拟前端模块部分。该部分内容包括模拟比较器、运算放大器模块、可编程 SC/CT 模块、温度传感器模块、基于混频器的精确整流实现等内容。

（14）ADC 和 DAC 模块部分。该部分的内容包括 Δ−Σ ADC 模块、DAC 模块、ADC 测量值显示的实现、IDAC 值显示的实现等内容。

（15）电容感应模块部分。该部分的内容包括电容感应模块的结构、电容感应算法、电容触摸感应实现等内容。

（16）数字滤波器模块部分。该部分的内容包括数字滤波器模块概述、数字滤波器模块结构、基于 DFB 的数字滤波器实现等内容。

（17）RTX51 Tiny 操作系统部分。该部分的内容包括 RTX51 Tiny 介绍、集成 RTX51 Tiny 到软件设计、程序结构及代码分析等内容。

为了让读者更好地掌握相关内容，本书每一章都给出了一个设计实例。

由于 PSoC 技术不断发展，其相应的设计资料也在不断更新中，读者可登录 www.cypress.com 网站下载最新数据手册。

本书不仅可以作为大学信息类专业的本科生、研究生的单片机、可编程片上系统相关课程的教学用书，也可以作为从事相关领域教学和科研工作者的参考用书。

本书由何宾编著，王纲领、常晓磊、彭渤也参与了编写工作。本书的编写，参考了 Cypress 公司最新的研究成果、设计文档等资料。Cypress 公司大学计划中国区经理魏荣博士为本书的编写提供了技术参考资料、PSoC3 硬件开发平台资源；Cypress 公司的技术工程师对本书编写过程中遇到的问题进行了耐心的回答。在此，对以上人员表示深深的谢意。

由于编者水平有限，编写时间仓促，书中难免有疏漏之处，敬请读者批评指正。

编著者

目 录

8051片上可编程系统原理及应用

第1章

PSoC 设计导论

Cypress 公司的可编程片上系统(Programmable System-on-Chip, PSoC)将微控制器、可编程逻辑阵列、模拟可编程阵列等资源集成在单芯片上，为电子系统的设计带来了前所未有的机遇。

本章主要介绍了微控制器基础、可编程片上系统 PSoC 概述、PSoC3 设计流程、PSoC3 的结构及功能和 PSoC3 器件概述。本章的内容是对 PSoC 所涉及知识的整体概述，通过本章内容的学习，帮助读者从“系统”角度把握 PSoC 技术的本质，方便对后续内容的学习和理解。

1.1 微控制器基础

微控制器是指带有外设的微处理器系统，比如台式电脑的 CPU，它是一个微处理器系统。微控制器将响应来自 I/O 引脚、定时器、通信等的输入，同时通过对信息进行操作控制来产生合适的输出信号。

I/O 引脚使得微控制器能读取来自其他设备的按钮和状态信息，同时 I/O 引脚也能够输出信号用来打开灯、运行电机和驱动显示设备。

定时器、通讯模块和数/模转换模块能使微控制器执行特殊的任务，比如与 PC 机进行通讯，读取温度信息等。

从微观上说，微控制器是一个集成了成千上万电子开关的设备。正如编程人员的目的是为了将复杂的操作简化为逻辑和算术运算来完成任务那样，微控制器的设计人员必须决定使用什么电子设备来完成这些任务，比如，晶体管，FET 和二极管等。大多数的微控制器工作在二进制系统下，比如，‘1’或‘0’，逻辑高或逻辑低，开或关。

Cypress 的微控制器系统称为可编程片上系统(Programmable System-on-Chip, PSoC)，那是因为在单芯片上包含了 CPU 内核、足够的模拟子系统和数字子系统资源。因此，在实现一个系统时，几乎不需要外部的电路。

1.1.1 微控制器的涵义

如图 1.1 所示，微处理器系统的 CPU 通常需要和其他部件相连接，这样才能使其发挥作用。微处理器系统通常会使用到的功能部件包括：

图 1.1 微处理器系统常用功能部件

① CPU：中央处理单元（Central Processing Unit, CPU）是系统的“大脑”，它知道如何和各种不同空间的存储器交换（读或写）信息。同时，也执行一些逻辑指令，最基本和最通

用的有：加、减、逻辑“或”、逻辑“与”、逻辑“异或”、移位、移动和复制。一些处理器可能执行更加复杂的操作，但这些操作都是由最基本的操作组合得到的。

CPU由一些子系统构成，在这些子系统中最重要的是程序计数器（Program Counter, PC），指令译码器和算术逻辑单元（Arithmetic Logic Unit，ALU）部分。PC指向Flash存储器指定的地址，然后返回指令和数据。PC用来确定送到指令译码器内的Flash中的字段。指令译码器包含译码逻辑，这些逻辑将对从Flash返回的数进行“翻译”，用来确定程序将执行的指令，这些指令将“告诉”CPU下一步所做的逻辑操作行为。

CPU不但能实现运算操作，也能修改程序运行的地址。如果在执行指令的过程中，并不是顺序的执行指令，比如遇到调转指令，那么PC将加载新的所要运行指令的地址，并且从指向Flash新的地址位置的地方执行程序。如果指令需要CPU执行一些运算，那么相关的数将送到ALU单元中。

此外，CPU也能根据所接收到的指令对外设进行控制。

② Cache：从位置和访问速度方面来说，高速缓存（Cache）最靠近CPU。有时，将高速缓存直接集成在同一芯片内。但并不是必须放在同一个硅片上，只是封装在同一个芯片内。

③ RAM：从CPU访问速度来说，访问随机访问存储器（Random Access Memory, RAM）比访问高速缓存要慢。需要说明的是，这个词语已经失去了它的原本含义，这是由于现在大部分的存储器都能够以任何顺序进行访问。

④ 硬件驱动：从速度来说，是系统中最慢和最大的存储部分。它用来保存程序，并且是由非易失性的存储介质构成。

1.1.2 微控制器编程语言

世界上不管是什么厂商的CPU，也不管它们采用什么样的结构，它们都有下面的共同特点：

- 都是靠程序计数器（Program Counter, PC）来控制程序的运行，正因为这个原因，它们本质上也是串行执行的；
- 工作在二进制状态下，也就是通常所说的在PC的控制下，通过运行二进制组成的机器代码，来控制CPU内各个功能部件的运行。

对CPU来说，所谓的“机器语言指令”，就是通过CPU内的控制逻辑来协调CPU内各个功能部件，完成所要求的操作。机器语言的运行效率是最高的。

机器语言指令应该由：操作码和操作数两部分构成。操作码告诉CPU所需要执行的操作，操作数是执行操作所针对的对象。这些对象包括：立即数、寄存器和存储器等，通过访问这些对象来获得所需要操作的对象。比如：对8051来说，机器语言指令-7D25，表示该指令要实现数据传输操作，“7D”是操作码，“25”是操作对象，其表示将十六进制数#25，送到R5寄存器中。

但是，正如前面所说的，纯粹意义上的“机器语言”对程序员太难理解了，为什么？这是因为程序员是CPU的操作者，而不是CPU的设计者，程序员根本不可能从二进制代码的排列中看出“机器语言”所描述的逻辑操作行为。

为了帮助程序员理解CPU所执行的操作，通过汇编语言助记符指令来帮助程序员设计指令来控制CPU的运行。汇编语言助记符指令通过汇编器被翻译成机器语言指令。用汇编助记符描述机器指令的形式为：

[标号：]　助记符　[操作数]　[；注释]

标号用来表示一行指令，助记符表示所要执行的逻辑操作行为，操作数为逻辑操作行为所操作具体对象，现在用汇编语言来描述上面的机器指令“7D25”：

MOV R5, #25

MOV 表示数据移动操作，R5 表示目的操作数，#25 表示源操作数，这个助记符汇编指令所表示的是，将立即数 25 复制到 R5 寄存器中。使用助记符来描述 CPU 所要进行的操作，比使用机器语言直接描述更加容易理解和记忆。但是，由于汇编语言下面是机器语言，所以对于使用汇编语言编程的程序员来说，他必须很清楚 CPU 的指令集，寄存器单元和存储器映射等繁琐的硬件规则。虽然其执行效率基本上和机器语言一样，但是使用汇编语言编程效率很低。因为很多程序员根本不了解 CPU 的具体内部结构，所以对它们来说，使用汇编语言编程并不比使用机器指令编程好到哪里去，这也是一件令他们非常痛苦的事情。

值得高兴的是，在今天，厂商开发的软件平台支持使用 C、BASIC 等高级语言对硬件进行编程；C 语言是不能直接在 CPU 上运行，它必须首先通过编译器（Compiler）转换成机器语言，才能在 CPU 上运行。使用高级语言所编写的代码其运行的效率不可能比用汇编语言编程的运行效率高，所以说，如果你想让 C 语言所编写的代码和汇编语言编写的代码有一样高的代码执行效率，你只能是绞尽脑汁的对 C 代码进行优化，或者使用 C 语言和汇编语言混合编程，来满足代码长度和运行时间的设计要求。虽然这一过程也会让程序员耗费很多的精力，但是，值得他们高兴的是，他们再也不用和底层硬件直接打交道了。

为了更清楚地说明它们之间的关系，通过表 1.1 来说明。

表 1.1 C 语言和汇编助记符之间的对应关系

C 语言描述	汇编助记符	机 器 指 令	功 能
F=C+D	MOV A, 0x08	E508	将数据空间地址为 0x08 的内容送给累加器 A
	ADD A, 0x09	2509	将数据空间地址为 0x09 的内容和累加器 A 相加后送给累加器 A
	MOV 0x0A, A	F50A	将累加器 A 的内容送到地址为 0x0A 的数据空间

现在更令程序员高兴的事情是，越来越多的厂商提供了硬件的应用程序接口（Application Program Interface，API）函数，这样程序员可以根本不用知道更多的硬件实现细节，只需关心如何编写代码来使硬件工作，这样就大大提高了程序的设计效率。

1.2 可编程片上系统 PSoC 概述

1.2.1 PSoC 发展概述

当今世界，嵌入式处理器无处不在。在过去三十年间，一方面，由于市场要求不断地降低嵌入式系统的成本；另一方面，要求嵌入式处理器处理及控制能力不断的提高；在这两个因素的推动下，使得嵌入式处理器的功能变得越来越复杂。所以，当半导体市场出现越来越多的片上可编程系统，设计者就不会感到奇怪了。这种 PSoC 其实质就是将 CPU、模拟和数字子系统集成在单芯片上。美国 Cypress 公司，率先在单芯片上实现完整的模拟和数字系统的集成，其典型的代表作即 PSoC1、PSoC3 和 PSoC5。

特别值得一提的是，PSoC3 和 PSoC5 片上可编程系统，由于在芯片内部分别集成了业界流行的 8051 CPU 硬核和 ARM Cortex-M3 CPU 硬核，使其受到业界的高度关注。以 PSoC 芯片和 PSoC Creator 软件开发工具为代表的硬件设计平台和软件设计工具，引领着未来嵌入式系统设计的发展方向。

其设计方法的核心就是，以不同的数字和模拟 IP 核“积木块”为中心的“系统级”设计，这种设计方法所体现出来的是对未来嵌入式设计者所要求的“重基础”和“宽专业”的要求。因为，这种 PSoC 具有很高的集成度，所涉及的知识内容也比较多，这就是“宽专业”，但是，

要想能完成PSoC的设计，要求设计者有非常好的基础理论知识，这就是“重基础”。

到此处，读者一定会提出这个问题，PSoC到底和传统8051有什么区别呢？作为新的嵌入式系统的设计平台，使用PSoC进行嵌入式系统设计具有以下几个方面的优点。

（1）定制

基于PSoC嵌入式系统的设计人员可以很灵活地选择所要连接的外设和控制器。因此，设计人员可以设计出一个独一无二的外设，这个外设可以直接和总线连接。对于一些非标准的外设，设计人员很容易得使用PSoC内嵌的通用数字块（Universal Digital Block，UDB）阵列实现对非标准外设的定制。比如，设计人员很容易在PSoC上设计出多个UART接口的嵌入式系统，而这些在传统的8051单片机和嵌入式系统是无法实现的。因此，在PSoC平台中，向这样类似的配置是很容易实现的。

（2）降低元件成本

由于基于PSoC平台的嵌入式系统的功能多样性，以前需要用很多元件才能实现的系统，现在可以使用一个PSoC芯片实现。比如，辅助I/O芯片或协处理器与现有的处理器之间的连接。减少在设计中所使用的元件的数量，不但可以降低元件的成本，而且可以大大缩小电路板的尺寸，提高系统的可靠性。

（3）硬件加速

选择PSoC的一个重要的原因就是，PSoC能在硬件和软件之间进行权衡，使嵌入式系统达到最大的效率和性能。比如，当算法是嵌入式系统软件性能的瓶颈时，一个使用定制的协处理器引擎能用来实现算法，这个协处理器通过专用的，低延迟的通道与嵌入式处理器连接。使用现代的硬件设计工具，很容易得将软件瓶颈转向硬件处理。

下面通过表1.2详细说明它们在软件和硬件设计上的区别。

表1.2 PSoC和传统8051的区别

项目	PSoC（8051内核）	传统8051
硬件	内部集成大量软件和硬件可配置的模拟模块	内部没有集成模拟模块
	内部根据用户要求可定制IP核	无用户定制IP核功能
	端口功能丰富，可配置模拟/数字功能、驱动能力等	端口功能单一
	引脚可以和内部模块任意连接	引脚固定，不能修改引脚和内部模块的连接方式
软件	汇编，C语言编程	汇编，C语言编程
	预建立/用户定制元件驱动函数自动生成	无元件驱动函数自动生成

综合上述，PSoC是一个MCU，但是是一个高度的硬件和软件可编程的MCU。在PSoC的平台上充满了设计的创意。

1.2.2 PSoC设计方法

1.2.2.1 设计背景

一个典型的嵌入式系统由下面三个主要模块构成：

- 处理器；
- 数字（外设和逻辑）；
- 模拟（用于与传感器和控制器的物理接口）；

系统根据硬件接口要求，由三个模块连接构成硬件平台。然后，在这个硬件设计上，“构建”软件，来执行所要求的处理和控制功能。

按照传统的设计方法，一旦定制的“软件”和“硬件”开始运行，对这个设计需要进行