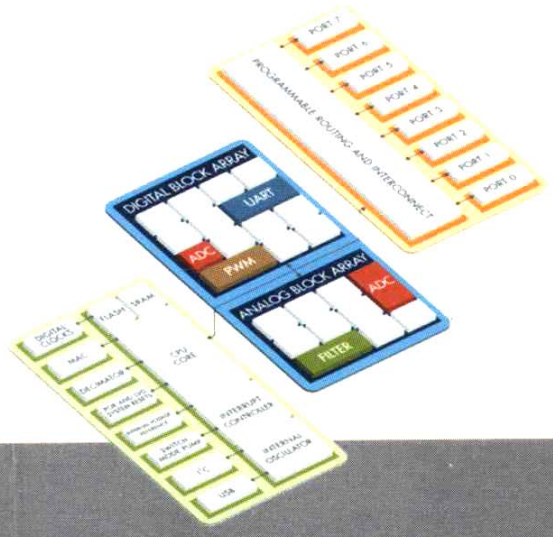
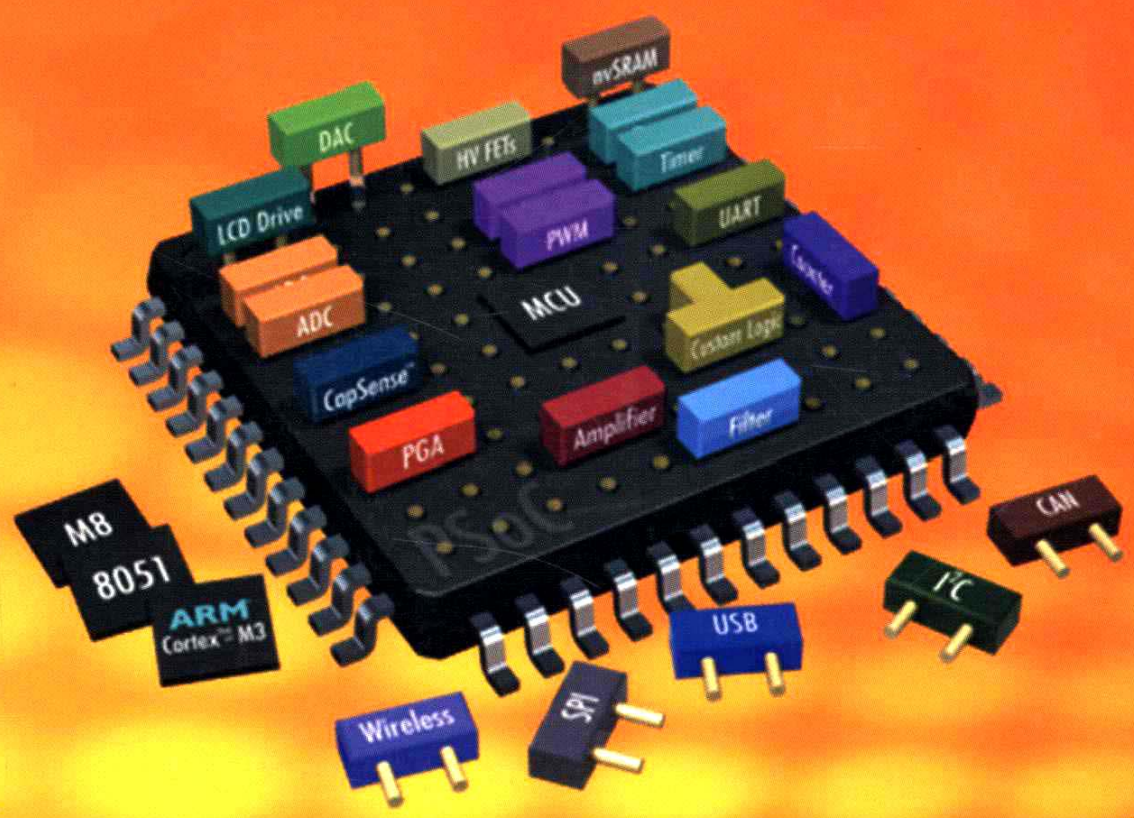
THE CYPRESS

CD-ROM
附赠光盘

可编程片上系统 PSoC 设计指南



Design Guide of Programmable System on chip



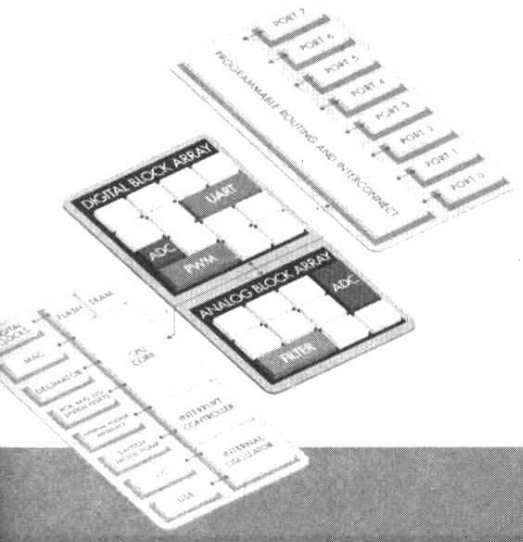
▶▶▶ 何 宾 编著

化学工业出版社

THE CYPRESS UNIVERSITY ALLIANCE PROGRAM

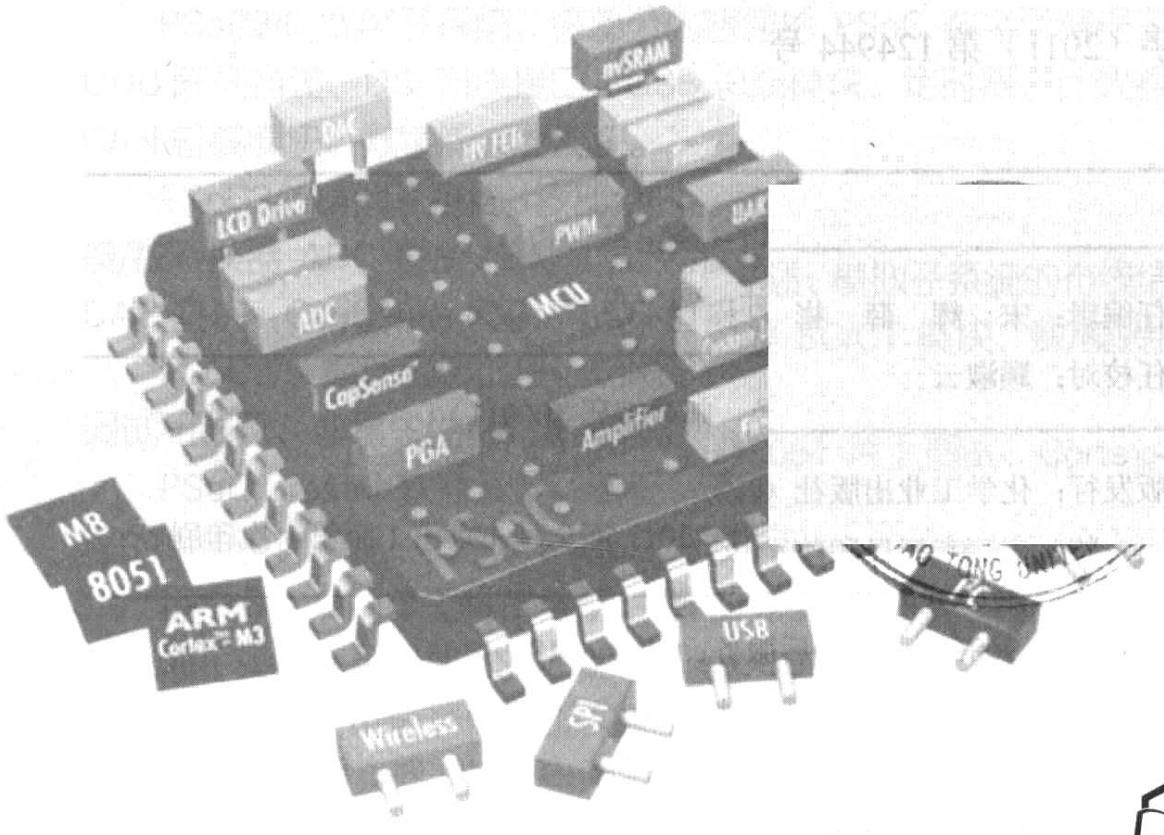
CD-ROM
附赠光盘

可编程片上系统 PSoC 设计指南



Design Guide of Programmable System on chip

▸▸▸ 何 宾 编著



化学工业出版社
·北 京·

图书在版编目（CIP）数据

可编程片上系统 PSoC 设计指南 / 何宾编著. —北京：化学工业出版社，2011.8

ISBN 978-7-122-11583-6

Ⅰ. 可… Ⅱ. 何… Ⅲ. 现场可编程门阵列-程序设计-指南 Ⅳ. TP331.2-62

中国版本图书馆 CIP 数据核字（2011）第 124944 号

责任编辑：宋 辉 薛 彬　　文字编辑：余纪军

责任校对：顾淑云　　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　　装：三河市延风印装厂

787mm×1092mm 1/16 印张 17¾ 插页 2 字数 468 千字　　2011 年 9 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：**48.00 元**

前　言

可编程片上系统 PSoC 设计指南

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。

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，PSoC5 集成了 ARM Cortex-M3 CPU 核。这种将传统嵌入式处理器集成到单芯片的结构，不但兼容了传统的嵌入式设计结构，而且将其从传统的应用领域扩展到了其它新的应用领域。

本书全面系统地介绍了 Cypress 公司的 PSoC3 和 PSoC5 可编程片上系统体系结构，主要包括以下几个部分。

PSoC 设计导论，该部分内容包括微控制器基础、可编程片上系统 PSoC 概述、PSoC3 器件概述、PSoC5 器件概述。

PSoC3/5CPU 子系统，该部分内容包括 8051CPU 核，Cortex-M3 CPU 核、Cache 控制器、DMA 和 PHUB 以及中断控制器。

PSoC3/5 存储器系统，该部分内容包括静态存储器 SRAM、Flash 程序存储器、EEPROM 模块、外部存储器接口 EMIF、存储器映射结构。

PSoC3/5 系统集成，该部分内容包括 PSoC 时钟管理、PSoC 的电源管理、PSoC 复位及 I/O 系统及布线。

PSoC3/5 数字子系统，该部分内容包括 PSoC 数字可编程子系统概述、通用数字块 UDB、UDB 阵列描述、DSI 布线接口、USB 总线模块、定时器、计数器和 PWM 模块、I^2C 总线模块、CAN 总线模块、数字滤波器模块。

PSoC3/5 模拟子系统，该部分内容包括 PSoC 模拟子系统功能概述、模拟子系统的布线结构、模/数转换器 ADC 模块、模拟比较器模块、运算放大器模块、可编程 SC/CT 模块、数/模转换器 DAC 模块、CapSense 模块、LCD 直接驱动模块、温度传感器模块。

PSoC3/5 编程和调试接口功能，该部分内容包括测试控制器、8051 片上调试、Cortex-M3 调试和跟踪、非易失性存储器编程。

PSoC Creator 软件及设计流程，该部分内容包括 PSoC Creator 的主要功能、基于 PSoC3 工程的简单设计流程、基于 PSoC5 工程的简单设计流程。

基于 PSoC 简单工程的实现，该部分内容包括 LED 显示控制的实现、LCD 显示 ADC 测试值的实现、正弦信号产生和显示的实现、USB 人体学输入设备的实现。

基于 PSoC 的信号传感的实现，该部分内容包括电容触摸感应实现、加速度传感器控制显示实现、水平仪的实现、热敏电阻测温的实现、接近度测量的实现。

基于 PSoC 的通信电路的实现，该部分内容包括压控振荡器 VCO 的实现、幅度调制 AM 的

实现、频率调制 FM 解调的实现。

为了让读者更好地掌握相关的内容，本书还配套光盘给出了大量示例程序。此外，在网上提供了 PPT 教学资源，读者可到 http://download.cip.com.cn 中的“配书资源”下载。

本书是可编程片上系统设计人员的自学、培训用书，也可以作为大学信息类专业讲授单片机、可编程片上系统相关课程的教学用书。

本书的编写得到了王纲领、常晓磊、何军和彭澎等人的帮助， Cypress 公司中国区大学计划部经理魏荣博士为本书的编写提供了数据书册、技术参考资料、PSoC3 和 PSoC5 硬件开发平台资源，Cypress 公司的 FAE 为本书的编写也提出了宝贵的意见，在此向各位朋友表示深深的谢意。

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

可编程片上系统 PSoC 设计指南

第1章

PSoC 设计导论

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 引脚使的微控制器能读取来自其它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。正如编程的人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 和 0，高和低，开和关。

Cypress 的微控制器系统称为可编程片上系统（Programmable System-on-a-Chip，PSoC），那是因为在单芯片上包含了足够的资源，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 通常需要和其它部件相连接，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 OR、逻辑 AND、逻辑 XOR、移位 Shift、移动 Move 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 存储器的指定地址，然后返回指令和数据。专用逻辑将使用 PC 来确定，Flash 中的将被送到指令译码器

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 下一步所将做什么。

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，并不是顺序的实行指令，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。

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 Cache 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）Hard Drive

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 数据和指令的处理

微控制器的指令译码器允许预定义指令集。在 PSoC1/3/5 中，有不同的指令集，所有的程序最后都要分解成这些预定义指令集中的指令。如果对 PSoC 使用 C 语言进行编程，C 语言编译器将 C 语言分解成这些预定义的指令。这些指令包含基本的逻辑和算术操作。这些指令中还有一些更复杂的指令，比如加、减、乘和比较操作。CPU 内包含这样的逻辑模块用来完成这些复杂的指令，而不需要将这些复杂的指令分解为简单的指令。

指令作为被编码的值保存在存储器中。编程人员不需要知道描述这些指令的值。汇编器接受助记符形式所描述的指令，这些助记符用来形式化的表示机器指令。这些指令有一个或多个操作数。操作码包含诸如算术逻辑操作、移位运算或控制程序执行顺序等操作类型。

通常情况下，第一个操作数为目的操作数，而其它操作数被认为是源操作数。如果指令用来改变程序执行的位置，那么操作数必须包含所要执行程序的新的地址。

表 1.1 给出 PSoC3（8051 核）C 语言和汇编助记符之间的对应关系。

表 1.1 C 语言和汇编助记符之间的对应关系

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

1.2 可编程片上系统 PSoC 概述

1.2.1 PSoC 技术特点

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，很容易将软件瓶颈转向硬件处理。

1.2.2　设计重用技术

很多年前，设计重用技术就已经提出来作为一个正在完成项目（从时间和预算方面）的一个必要的部分。这并不是一个新的思想，设计重用的目的就是使得不需要修改一个设计（或者尽可能少的修改），就可以使该设计可以在不同的平台之间运行和实现。实际上，到目前为止，设计重用都没有完全实现，通常的做法是单纯的硬件（微处理器核，可重用的 IP 核外设，硬件加速器等）或者软件（RTOS，协议栈，实时库等等）的复用，而不是全部软件和硬件的复用。

设计重用的思想非常吸引人，但是目前的标准的解决方式仅仅是迫使在项目的另一部分加入“定制”的开发，这不能根本上解决问题。

使得 IP 核重用利益最大化的方法是将软件和硬件作为“同等地位”的“合作者”，不需要使得软件或硬件的任何一方需要了解对方的具体实现过程。实现这个目的的方法是，在定义硬件 IP 核的时候，顺便也要考虑到软件和开发工具，这样在应用程序和硬件之间的接口就非常方便、高效，同时，彼此不需要“深入了解对方”。

Cypress 的 PSoC Creator 开发平台很好地实现了设计重用的思想，并将其变为现实，即在生成硬件 IP 时，也同时提供了相应所需要的软件 API 函数，这样使得设计更容易运行，以更快的速度完成，更加容易维护和便携。

当使用 HDL 语言开发 IP 核时，对其进行综合、仿真、验证、编写测试平台、编写文档。那么应该为 IP 核的使用者在 IP 核开发工具中提供相同的工具，这个工具就是当用户在他的设计中例化所需要使用的 IP 核时，为每个例化的 IP 核生成相应的 API 函数。

在 PSoC Creator 中支持这样的功能。PSoC3/5 由嵌入式的处理器（8051 或 Cortex-M3）、可编程的数字阵列和高精度的模拟资源构成。PSoC Creator 软件充分的显示出其强大的设计重用功能，即硬件模块或者元件能用 API 函数进行封装来简化软件的开发，同时加速设计

过程。

PSoC Creator 提供原理图捕获接口，在原理图界面内，设计者通过从模拟和数字元件库中拖拽元件来创建设计。一个元件由一个在原理图界面内可见的符号或者其它原理图的实现构成。当设计者建立（build）设计时，软件就根据元件的名字产生相应的 API 函数。

PSoC Creator 内的元件都是参数化的，那些重要的设置选项，比如 UART 中的流控制和波特率，通过参数化图形设置很容易将那些不必要的功能根据设置选项从具体实现中删除。

下面给出一个例子来说明 IP 核设计和软件 API 函数的“定制”。如图 1.2 所示，使用 PSoC Creator 软件工具创建 UART_1 和 UART_2 两个例化的 IP 核。UART_1 支持 Tx 和 Rx，中断模式，硬件的发送使能信号 tx_en 信号（用于 RS-485）；而 UART_2 是一个简单的实现，即在轮询方式下输出数据流。在后一种情况下，PSoC Creator 不创建用于读数据、中断，检查 Rx 状态和缓冲区管理的 API 函数，这样可以减少混乱和错误的产生。

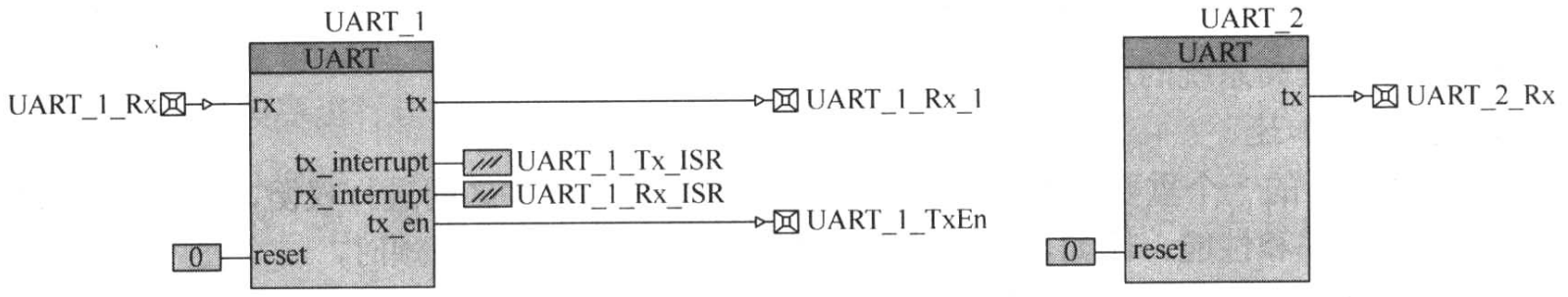


图 1.2　API 函数的定制

通过参数化的设置，PSoC Creator 也支持交叉结构。比如在 PSoC5 的 UART 有较大的缓冲区，并且从 GetBufferSize()中返回值，比如可选择是一个字符或者一个短值（8 位 1～255，32 位 1～65535），同时保证其在 8051 结构中是有效的。如图 1.3 所示，通过使用宏定义中断例程，允许在任何编译器中使用相同的代码。在系统的头文件中定义了宏，比如在 8051 编译器中（ARM 的 Keil）识别定义“_C51_”，并且将中断关键字添加到函数定义中。对于 ARM GNU 编译器（CodeSourcery），“_GNUC_”定义保证删除不必要的关键字。

cytypes.h

```
#define CY_ISR(FuncName)   void FuncName (void)
#elif ((defined(__C51__))
#define CY_ISR(FuncName)   void FuncName (void) interrupt 0
#endif
```

UART_1_INT.c

```
CY_ISR(UART_1_TXISR)
{
    while( !(UART_1_TXSTATUS & UART_1_TX_STS_FIFO_FULL) )
    {
        ...
```

图 1.3　中断代码的重用

1.3 PSoC3 器件概述

1.3.1 PSoC3 功能和特点

如图 1.4 所示，PSoC3 包含的主要部件有：

- ✓ 8051 CPU 子系统；
- ✓ 非易失性存储子系统；

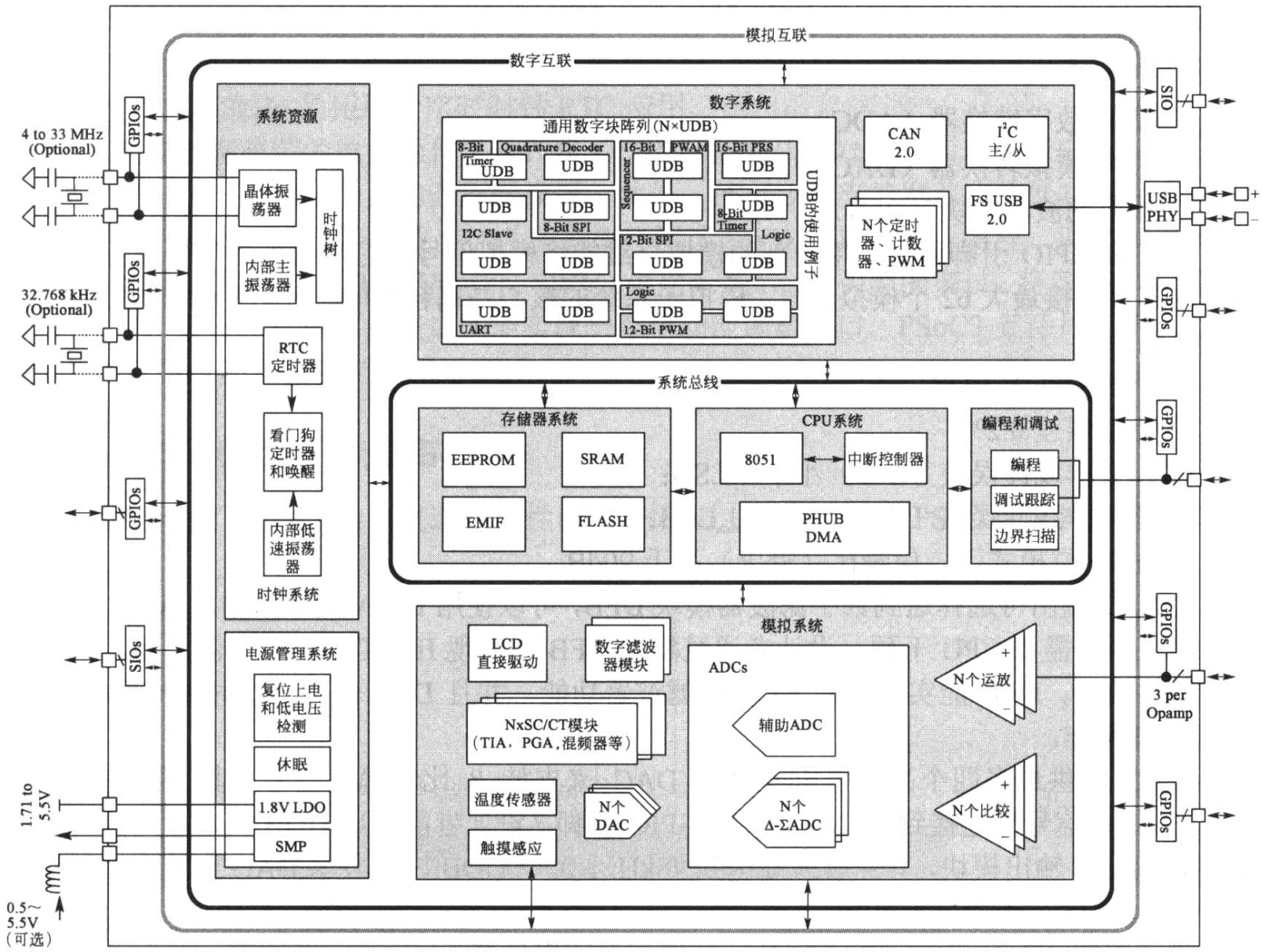


图 1.4　PSoC3 的结构原理

✓ 编程，调试和测试子系统；
✓ 输入和输出；
✓ 时钟资源；
✓ 电源；
✓ 数字子系统；
✓ 模拟子系统；

数字子系统为整个 PSoC3 系统提供了一半的可配置能力。通过数字系统互联（Digital System Interconnect，DSI）将来自外设的数字信号连接到任意引脚。它通过小的、快速的、低功耗的 UDB 阵列提供了灵活的功能。PSoC Creator 提供了预建立和测试的标准数字外设库（UART，SPI，LIN，PRS，CRC，定时器，计数器，PWM，AND，OR 等），并将其映射到 UDB 阵列上。设计者可以很容易的通过在图形化的设计界面中使用布尔原语函数来创建数字电路。每个 UDB 包含可编程的阵列逻辑（Programmable Array Logic，PAL）和可编程逻辑设备（Programmable Logic Device，PLD）。同时，和小规模的状态机引擎一起支持不同的外设。

除了提供灵活的 UDB 阵列外，PSoC 也提供了可配置的数字块用于指定的功能。对于 PSoC3 这些块包含最大四个 16 位的定时器，计数器和 PWM 块；I^2C 从设备、主设备和多主设备；全速 USB 和 CAN2.0b。

模拟子系统为整个 PSoC3 系统提供了另一半的可配置能力。所有的模拟性能基于高精度的绝对电压参考。配置模拟子系统包括：

✓ 模拟交叉开关；

✓ 比较器；
✓ 电压参考；
✓ 模拟-数字转换器（ADC）；
✓ 数字-模拟转换器（DAC）；
✓ 数字滤波器模块（DFB）。

所有的 GPIO 引脚通过使用内部的模拟总线能将模拟信号和芯片内外的信号进行连接。这允许芯片连接最大 62 个模拟信号。模拟子系统的核心是高精度的、可配置的 Σ-ΔADC，其特征包括：

✓ 小于 100μV 偏置；
✓ 增益误差 0.2%；
✓ 积分非线性误差（INL）小于 1 LSB；
✓ 差分非线性误差（DNL）小于 1 LSB；
✓ 在 16 位模式下，信噪比（SNR）高于 90dB。

ADC 的输出可选择送到数字滤波器模块 DFB，可以使用 DMA 完成数据从 ADC 到 DFB 的传输，而不需要 CPU 干预。设计者通过配置 DFB 来实现 IIR 和 FIR 数字滤波器，或者用户定制的功能。DFB 能实现最多 64 阶的滤波器功能，并且 DFB 能在一个时钟周期执行 48 位乘-累加操作。

PSoC 提供最多四个高速电压或电流 DAC 来支持 8 比特输出信号，其最高速度达到 8Msps。这些信号能连接到芯片上任意的 GPIO 引脚。设计者能使用 UDB 创建更高分辨率电压 PWM DAC 输出模块，其速度最高达到 48kHz。每个 UDB 内的数字 DAC 支持可编程宽度的 PWM，PRS，或者 Σ-Δ 算法。

除了支持 ADC，DAC 和 DFB 外，模拟子系统提供多个：

- 可供使用的运算放大器；
- 可配置的开关电容（Switched Capacitor，SC）/连续时间（Continuous Time，CT）块。

这些块支持：互阻放大器（电流-电压）、可编程增益放大器、混频器和其它类似的模拟元件。

PSoC3 的 8051 CPU 子系统使用单周期的流水结构的 8051 微处理器，最高运行速度达到 67MHz。CPU 子系统包括可编程嵌套的中断控制器，DMA 控制器和 RAM。PSoC3 内的单周期运行的 8051 子系统比标准的 8051 处理器速度快几十倍。

PSoC3 的非易失性子系统由 Flash，字节宽度可写的 EEPROM 和非易失性的配置选项构成。PSoC3 提供最大 64KB 存储容量的片上 Flash。CPU 能对 Flash 的每个块重新编程和使用启动代码。设计者可以使用纠错码 ECC，使得 PSoC3 应用于高可靠性的领域。PSoC3 提供了强大的安全保护措施，用于阻止对 PSoC3 上非易失性存储器系统进行非法的读写操作。PSoC3 上提供了最大 2KB 可写的 EEPROM，可以用于保存应用数据。

PSoC 上提供了 3 种类型的 I/O，其使用非常灵活。在 POR 的时候，所有的 I/O 可以设置多种驱动模式。PSoC3 通过 V_{ddio} 引脚，提供最多四个 I/O 电压域。每个 GPIO 都有模拟 I/O，LCD 驱动，电容感应，产生中断，抖动率控制和数字 I/O 的能力。当作为输出时，PSoC 上的 SIO 允许独立于 V_{ddio} 设置 V_{oh}。当 SIO 为输入模式时，输出为高阻状态。SIO 非常适合使用 I^2C 总线的应用（即当总线上有其它设备，而 PSoC 又没有上电的情况）。SIO 也提供大电流驱动能力用来驱动 LED。SIO 可编程的输入门槛特性能使 SIO 作为通用的模拟比较器。对于全速的 USB，USB 接口也提供了 USBIO。当不使用 USB 时，这些引脚被用于有限的数字功能和对芯片进行编程。

PSoC3 提供了灵活的内部时钟生成器。内部主时钟振荡器 IMO 是系统的主时钟基准（3MHz 的精度为±1%），IMO 的频率范围 3～62MHz。从主时钟频率产生多个衍生时钟来满