高等学校应用型本科“十三五”规划教材

机 电 MECHATRONICS

计算机 COMPUTER

电 子 ELECTRONICS

- 理论与应用结合紧密，实例丰富
- 基于μC/OS-II介绍关键技术

# 嵌入式操作系统及ARM Cortex-M0+应用

张勇 安鹏 编著

西安电子科技大学出版社
http://www.xduph.com

高等学校应用型本科“十三五”规划教材

# 嵌入式操作系统及ARM Cortex-M0+应用

张 勇 安 鹏 编著

西安电子科技大学出版社

## 内容简介

本书基于 μC/OS－Ⅱ和 ARM Cortex-M0＋内核微控制器 LPC824 详细讲述了嵌入式实时操作系统的应用开发技术，主要内容包括 ARM Cortex-M0＋内核、LPC82X 微控制器、CPC824 开发平台与工程框架、异常与中断管理、μC/OS－Ⅱ工作原理及其移植、μC/OS－Ⅱ任务、μC/OS－Ⅱ信号量与互斥信号量、μC/OS－Ⅱ消息邮箱与队列、μC/OS－Ⅱ高级系统组件、LPC82X 典型应用实例等。

本书的特色在于理论与应用紧密结合，实例丰富，对学习嵌入式实时操作系统 μC/OS－Ⅱ及其在 Cortex-M0＋微控制器方面的教学与工程应用，都具有一定的指导和参考价值。

本书可作为普通高等院校电子信息、通信工程、计算机工程、软件工程、自动控制、智能仪器和物联网等相关专业的高年级本科生教材，也可作为嵌入式系统爱好者和工程开发人员的参考用书。

**图书在版编目(CIP)数据**

**嵌入式操作系统及 ARM Cortex-M0＋应用**/张勇，安鹏编著.
—西安：西安电子科技大学出版社，2015.8
高等学校应用型本科“十三五”规划教材
ISBN 978－7－5606－3772－3

Ⅰ. ① 嵌… Ⅱ. ① 张… ② 安… Ⅲ. ① 微处理器—系统设计—高等学校—教材
Ⅳ. ① TP332

**中国版本图书馆 CIP 数据核字(2015)第 157847 号**

策划编辑　李惠萍
责任编辑　阎　彬　郭　魁
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　北京京华虎彩印刷有限公司
版　　次　2015 年 8 月第 1 版　2015 年 8 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印张　20
字　　数　468 千字
定　　价　35.00 元
ISBN 978－5606－3772－3/TP
**XDUP　4064001－1**

# 西安电子科技大学出版社
# 高等学校应用型本科“十三五”规划教材

# 编审专家委员名单

主　任：鲍吉龙（宁波工程学院副院长、教授）

副主任：彭　军（重庆科技学院电气与信息工程学院院长、教授）

张国云（湖南理工学院信息与通信工程学院院长、教授）

刘黎明（南阳理工学院软件学院院长、教授）

庞兴华（南阳理工学院机械与汽车工程学院副院长、教授）

**电子与通信组**

**组　长：**彭　军（兼）

张国云（兼）

成　员：（成员按姓氏笔画排列）

王天宝（成都信息工程学院通信学院院长、教授）

安　鹏（宁波工程学院电子与信息工程学院副院长、副教授）

朱清慧（南阳理工学院电子与电气工程学院副院长、教授）

沈汉鑫（厦门理工学院光电与通信工程学院副院长、副教授）

苏世栋（运城学院物理与电子工程系副主任、副教授）

杨光松（集美大学信息工程学院副院长、教授）

钮王杰（运城学院机电工程系副主任、副教授）

唐德东（重庆科技学院电气与信息工程学院副院长、教授）

谢　东（重庆科技学院电气与信息工程学院自动化系主任、教授）

楼建明（宁波工程学院电子与信息工程学院副院长、副教授）

湛腾西（湖南理工学院信息与通信工程学院教授）

**计算机大组**

**组　长：**刘黎明(兼)

成　员：(成员按姓氏笔画排列)

刘克成(南阳理工学院计算机学院院长、教授)

毕如田(山西农业大学资源环境学院副院长、教授)

李富忠(山西农业大学软件学院院长、教授)

向　毅(重庆科技学院电气与信息工程学院院长助理、教授)

张晓民(南阳理工学院软件学院副院长、副教授)

何明星(西华大学数学与计算机学院院长、教授)

范剑波(宁波工程学院理学院副院长、教授)

赵润林(山西运城学院计算机科学与技术系副主任、副教授)

雷　亮(重庆科技学院电气与信息工程学院计算机系主任、副教授)

黑新宏(西安理工大学计算机学院副院长、教授)

# 前　言

传统的8051系列单片机由于具有硬件结构简单、编程操作方便以及芯片价格低廉等特点，长期以来被广泛应用于智能控制和显示等嵌入式系统中。此外，单片机的典型开源工程项目和优秀教材资源丰富，目前在普通高等院校中，几乎所有的电子工程和智能控制相关专业都设有单片机课程。随着科技的发展和人们对高智能性控制设备的喜爱与需求，传统单片机因其控制逻辑简单而在很多领域显得应用乏力。因此，近些年来，很多半导体公司推出了兼容8051系统传统单片机的新型增强型单片机，例如Silicon Labs公司的C8051F系列模数混合型单片机、Atmel公司的megaAVR单片机、Renesas公司的RL78系列单片机和TI公司的MSP430系列单片机等。新型单片机具有存储空间大、代码效率高、片上外设丰富和执行速度快等优点，在一定程度上延缓了单片机的应用衰退趋势，但无法从根本上改变单片机正在慢慢退出嵌入式应用系统的趋势。

当前，ARM微控制器正在逐步替代传统单片机而成为嵌入式系统的核心控制器。ARM公司出品了众多微处理器内核，包括目前市场上流行的ARM7、ARM9和ARM11内核。2010年以后，ARM公司主推的内核为Cortex系列。这个系列又分为M系、R系和A系。其中，A系是高性能系列，针对带有Android操作系统的智能平板电脑，支持ARM、Thumb和Thumb-2指令集；R系为普通嵌入式内核，支持ARM、Thumb和Thumb-2指令集；M系为低功耗系列，仅支持Thumb-2指令集，诞生于2004年，最早推出的内核为Cortex-M3，目前有Cortex-M0、M0＋、M1、M3、M4和M7，用于需要快速中断的嵌入式实时应用系统中。在Cortex系列中，M系列芯片的应用量最大，截至2014年中期，Cortex-M系列内核的微控制器的应用量达80亿颗，超过其他所有ARM内核的芯片用量的总和。

在Cortex-M系列中，最早推出的M3内核主要针对控制领域中的高端实时应用领域，具有控制和数字信号处理能力，除了可用于传统8051单片机的应用领域外，还可用于DSP处理器的应用领域，代表芯片有NXP(恩智浦)公司的LPC1788微控制器，该系列微控制器还被用作苹果iPhone手机中的协处理器；M4内核主要针对高速控制、语音信号处理和数字信号处理领域，涵盖了传统网络控制芯片和DSP处理器的应用领域，代表芯片有NXP公司的LPC4088微控制器；M7内核是2014年新推出的低功耗高性能内核，主要针对物联网、智能家居和可穿戴设备，具有音频和视频处理能力，代表芯片有意法半导体公司的STM32F7系列微处理器；M0和M0＋内核都是低功耗内核，M0＋内核的功耗比M0内核更低(ARM公司公布的功耗数据为11.2 μW/MHz)，被誉为全球功耗最低的微控制器内核，主要应用在控制和检测领域，涵盖了传统8051单片机的应用领域，比传统8051单片机在处理速度、功耗、片上外设灵活多样性、中断数量与中断反应能力、编程与调试等诸多方面都有更大优势，M0＋内核的代表芯片有NXP公司的LPC824微控制器。

基于微控制器的软件开发有两种方式，即无操作系统的应用软件开发和加载嵌入式实时操作系统的应用软件开发。前者称为函数级程序设计方法(或面向函数的程序设计方

法)，后者称为任务级程序设计方法(或面向任务的程序设计方法)。由于传统 8051 单片机片上的 RAM(随机访问存储器，可读可写)空间有限，一般在 4 KB 以内，不适宜加载嵌入式操作系统，因此，常借助于汇编语言或 C51 语言编写实现特定功能的函数，这些函数通过互相循环调用或外部中断触发调用的方式依次执行。这种函数级的程序设计的致命缺点在于当实现的功能较复杂时，无法保证多个功能单元的同步执行。例如，某个功能单元设计要求为严格地每隔一秒执行一次，但是在函数级的工程程序中，总会出现该功能单元间隔 1.01 秒、0.99 秒或 0.995 秒依次执行的情况，有时相邻两次间隔也不相等，例如分别为1.001 秒和 0.997 秒等，而且无法从根本上解决这个问题。

ARM 微控制器由于片内 RAM 空间丰富，一般在 8 KB 以上，适宜加载嵌入式实时操作系统(RTOS)。常用的 RTOS 有 μC/OS－Ⅱ、μC/OS－Ⅲ、FreeRTOS 和嵌入式 Linux，这些 RTOS 都是开放源代码的操作系统。针对 Cortex-M0＋内核而言，作者更偏爱 Micrium 公司的 μC/OS－Ⅱ和 μC/OS－Ⅲ。在 ARM 微控制器上加载了 RTOS 后，通过 C 语言编写实现各个特定功能的用户任务(而不是函数，任务可调用函数)，由 RTOS 管理和调度各个用户任务，实现各个功能单元的同步执行。

一般地，函数级程序设计要求程序员对硬件资源和外设接口非常熟悉，需要根据硬件时序定义编写这些硬件单元的驱动程序，还要编写实现用户功能的程序代码；而任务级程序设计方法简化了程序员的设计工作，在硬件平台上移植了 RTOS 后，RTOS 开发商会提供所谓的板级支持包(BSP)，其功能相当于计算机的显卡、声卡和网卡等的驱动。通过板级支持包，程序员可以以调用函数的形式访问微控制器的片上外设资源，这样软件开发程序员只需要专注于实现各个功能单元的程序代码，而无需深入了解硬件资源，甚至不需要懂得硬件工作原理，这大大加快了项目的开发进度。

希望上述内容能够回答很多读者关于“为什么要学习 ARM Cortex-M0＋内核微控制器”和“为什么要学习嵌入式实时操作系统”等问题。

本书将阐述基于 Cortex-M0＋内核的 LPC824 微控制器和嵌入式实时操作系统 μC/OS－Ⅱ的系统应用和程序设计方法。由于 LPC824 具有开关矩阵外设(Switch Matrix，也被译为端口配置矩阵单元)，因此 LPC824 在硬件电路设计上特别灵活，在产品升级换代时，只需要通过软件编程方式修改端口配置矩阵，而不需要重新设计电路板(类似于 FPGA 芯片)。并且，LPC824 还具有编程方便、处理速度快和控制能力强等特点，有些专家称 LPC824 是具有划时代标志特征的微控制器芯片。嵌入式实时操作系统 μC/OS－Ⅱ是美国 Micrium 公司推出的微内核，其最新版本为 v2.91，最多可以支持 255 个任务，具有实时性强、中断处理速度快和内核体积小等显著优点，特别适合移植到以 ARM Cortex-M 系列微控制器为核心的硬件系统平台上。

## 一、本书内容介绍

本书分为三篇，共十三章。江西财经大学软件与通信工程学院张勇编写了第一至五章和第十至十三章；宁波工程学院电子与信息学院安鹏编写了第六至九章。全书内容概括如下：

第一篇(一至四章)为全书的硬件基础部分，依次介绍了 ARM Cortex-M0＋内核、LPC82X 微控制器、LPC824 开发平台与工程框架以及异常与中断管理。

第二篇(五至九章)详细介绍了基于嵌入式实时操作系统 μC/OS－Ⅱ的任务级别的程序

设计方法，依次讲述了 μC/OS－Ⅱ工作原理及其移植、μC/OS－Ⅱ任务、μC/OS－Ⅱ信号量与互斥信号量、μC/OS－Ⅱ消息邮箱与队列和 μC/OS－Ⅱ高级系统组件。这部分内容结合了具体的工程实例，并给出了拓展思考题。

第三篇(十至十三章)为典型应用实例，依次介绍了基于 LPC824 学习板的智能门密码锁设计实例，智能温度采集、显示与报警系统以及数字电压采集与显示实例，最后给出了一个 NXP 公司设计的开源硬件平台 LPCXpresso824-MAX。该学习平台硬件和软件设计规范，且学习资料丰富，可作为基于 LPC824 设计应用系统的开发模板。这里所谓的“开源硬件”，是指硬件原理图和 PCB(印刷电路板)图均公开的硬件平台，且这类硬件平台易于学习并可实现二次开发，众所周知的 Arduino 系列硬件均为代表性的开源硬件。

**二、本书教学思路**

本书初稿已经过多名教师教学应用，理论课时宜为 32～48 学时，实验课时为 32 学时，开放实验课时为 16 学时。建议讲述内容为第一至九章，并按书中章节顺序讲述；第十至十二章用于课程设计和学生实验拓展。针对教师教学活动，作者提供了更多的交流和技术支持，可通过西安电子科技大学出版社或信箱 zhangyong@jxufe.edu.cn 与我们联系。

建议理论教学与实验教学同步进行。理论教学过程中，可设置 2～4 学时讨论课，或安排学生分组作学习交流主题报告。实验教学可设置 3～4 个基础性实验和 1～2 个综合性或设计性实验，可结合全国大学生电子设计大赛或嵌入式系统大赛开展拓展性实验项目，并应以学生动手为主。

对于自学本书的嵌入式系统爱好者而言，要求至少具有数字电路、模拟电路、C 语言程序设计等课程的基础知识，并建议在学习过程中设计一套 LPC824 学习板配套学习。

本书的每个实例都是完整的，读者可以在西安电子科技大学出版社网站上下载到全部工程实例代码，但作者强烈建议读者自行输入实例代码，以增强学习与记忆效果。

**三、本书特色**

本书具有以下四个方面的特色：

其一，详细讲述了基于 ARM Cortex-M0＋内核 LPC824 微控制器为核心的开源硬件平台，该平台包括了典型的 LED 灯、串口、按键、蜂鸣器、数码管、JTAG(SWD)和 ISP 电路、测温电路、模数转换电路和 LCD 屏电路等，对嵌入式硬件开发具有一定的指导作用。

其二，全书工程实例丰富，通过完整的工程实例详细讲述了函数级别与任务级别的程序设计方法，对于嵌入式系统应用软件开发具有较强的指导作用。

其三，结合 LPC824 硬件平台，详细讲述了嵌入式实时操作系统 μC/OS－Ⅱ的任务管理和系统组件应用方法，对学习和应用 μC/OS－Ⅱ具有较好的可借鉴性。

其四，通过典型的项目应用实例，详细讲述了嵌入式系统的软件开发与设计技术，对嵌入式系统项目开发具有一定的指导意义。

**四、致谢**

感谢 NXP 公司为本书编写提供了学习板和集成开发环境。在本书编写过程中，NXP 公司的辛华峰、王朋朋和张宇等专家提供了大量技术支持，并阅读了本书初稿，提出了很多建设性意见。

感谢北京博创兴盛陆海军总经理、广州天嵌科技梁传智总经理、北京赛佰特张方杰总经理、北京麦克泰公司曹旭华总经理对本书出版的关心和支持。

还要特别感谢阅读了作者已出版的图书并反馈了宝贵意见的读者们，他们使得本书的写作能按照“认识—应用—拓展”的思路进行，从而使得自学门槛较以前出版的书大为降低。

最后感谢西安电子科技大学出版社李惠萍编辑为本书出版所付出的辛勤工作。

由于作者水平有限，书中难免会有纰漏之处，敬请同行专家和读者批评指正。

**五、免责声明**

知识的发展和科技的进步是多元的。本书内容广泛引用的知识点大都出自相关文献，主要为 LPC824 用户手册，LPC824 芯片手册，Cortex-M0＋技术手册，嵌入式实时操作系统 μC/OS－Ⅱ、Keil MDK 集成开发环境和 ULINK2 仿真资料等内容，所有这些被引用内容的知识产权归相关公司所有，作者保留其余内容的所有权利。本书内容仅用于教学目的，旨在推广 ARM Cortex－M0＋内核 LPC824 微控制器、嵌入式实时操作系统 μC/OS-Ⅱ和 Keil MDK 集成开发环境等，禁止任何单位和个人摘抄或扩充本书内容用于出版发行，严禁将本书内容用于商业场合。

作　者

2015 年 7 月于

江西财经大学枫林园

# 目　录

## 第一篇　LPC82X 典型硬件系统

## 第二篇 嵌入式实时操作系统 μC/OS-Ⅱ的应用

## 第三篇 LPC82X 典型应用实例

# 第一篇

# LPC82X 典型硬件系统

本篇包括第一至四章，为全书的硬件基础部分，依次介绍了 ARM Cortex-M0＋内核、LPC82X 微控器、LPC824 开发平台与工程框架以及异常与中断管理等内容。

# 第一章 ARM Cortex-M0＋内核

ARM 是 Advanced RISC Machine(高级精简指令集机器)的缩写，现为 ARM 公司的注册商标。ARM Cortex-M0＋内核属于 ARM 公司推出的 Cortex-M 系列内核之一，相对于高性能的 Cortex-M3 内核而言，它具有体积小、功耗低和控制灵活等特点，主要针对传统单片机的控制与显示等嵌入式系统应用。本章将介绍 Cortex-M0＋内核的特点、架构、存储器配置和内核寄存器等内容。

## 1.1 ARM Cortex-M0＋内核特点

Cortex-M0＋内核使用 ARMv6-M 体系结构，使用 ARMv6-M 汇编语言指令集，它具有以下特点：

(1) Cortex-M0＋内核包含极低数量的门电路，是目前全球功耗最低的内核，特别适用于对功耗要求苛刻的嵌入式系统应用场合。

(2) 支持 32 位长的 Thumb-2 扩展指令集和 16 位长的 Thumb 指令集，代码的执行效率远远高于 8 位长的单片机汇编指令。

(3) 支持单周期的 I/O(输入/输出)口访问，对外设的控制速度快。

(4) 具有低功耗工作模式，在内核空闲时可以使其进入低功耗模式，从而极大地节约电能；当内核要工作时，通过紧耦合的快速中断唤醒单元使其进入正常工作模式。

(5) 内核中的各个组件采用模块化结构，通过精简的高性能总线(AHB-Lite)连接在一起，内核中的功耗管理单元可以动态配置各个组件的工作状态，可根据需要使某些空闲的组件处于掉电模式，以尽可能地减少功耗。

(6) 可执行代码保存在 Flash 存储区中，而 Cortex-M0＋内核支持从 Flash 中以极低的功耗快速读取指令，并以极低的功耗(工作在一个相对较低的 CPU 时钟下)在内核中高速执行代码。

(7) Cortex-M0＋内核支持硬件乘法器。硬件乘法器最早出现在 DSP(数字信号处理器)芯片中，与加法器协同工作，并称为乘加器，是指在一个 CPU 时钟周期内，用硬件电路直接实现 A×B＋C 的三操作数运算。这里 Cortex-M0＋支持的硬件乘法器可以在一个 CPU 时钟周期内实现 A×B 的二操作数运算。

(8) Cortex-M0＋内核的每条汇编指令的执行周期是确定的，中断处理的时间是确定的、高效的，且具有快速中断处理能力，特别适用于对实时性要求苛刻的智能控制场合。

(9)支持二线的串行调试接口(SWD)，只需要使用芯片的两根管脚就可以实现对 Cortex-M0＋内核芯片的在线仿真与调试，通过 SWD 可以向芯片的 Flash 存储器固化程序代码，且具有指令跟踪执行功能。而绝大多数的传统单片机是不能在线仿真调试的，因此，基于单片机的工程测试复杂且周期漫长。

(10) Cortex-M0＋内核不是物理形态的微控制器芯片，而是属于知识产权(IP)，所以常被称为IP核。目前全球大约有150家半导体公司购买了ARM公司的IP核，生产集成了IP核的微控制器芯片(称为流片，流片测试成功后进入芯片量产阶段)。所有集成了Cortex-M0＋内核的微控制器芯片，均可使用相同的集成开发环境(如Keil公司的MDK和IAR公司的EWARM等)和相同的仿真器(如ULink2、JLink V8等)进行软件开发，事实上，几乎全部的ARM芯片都使用相同的开发环境和仿真器。但是，对于传统的单片机而言，不同半导体厂商生产的单片机所用的开发环境和编程下载器往往不相同。

## 1.2　ARM Cortex-M0＋内核架构

相对于8位字长的传统8051单片机而言，Cortex-M0＋内核是32位字长的微控制器内核，其内部总线宽度为32位，指令和数据传输速度及功能大大提升。Cortex-M0＋内核架构如图1－1所示。

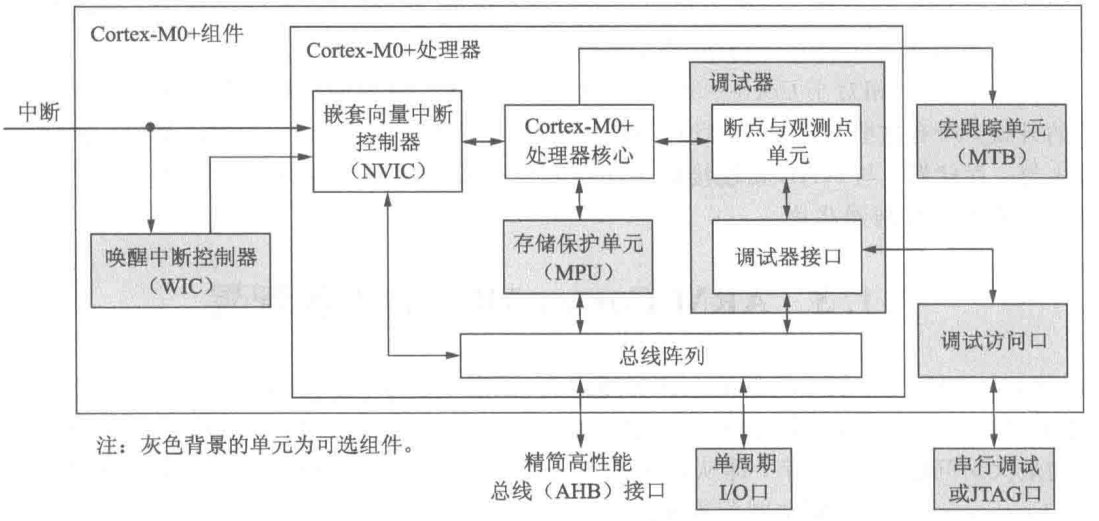


图1－1　Cortex-M0＋内核架构

由图1－1可知，Cortex-M0＋内核由Cortex-M0＋处理器和三个可选的组件，即唤醒中断控制器(WIC)、宏跟踪单元(MTB)和调试访问口组成，Cortex-M0＋处理器包括Cortex-M0＋处理器核心、嵌套向量中断控制器(NVIC)和两个可选的组件，即存储保护单元(MPU)和调试器组成，其中，调试器又包括断点与观测点单元和调试器接口。Cortex-M0＋内核与外部通过总线阵列和中断进行通信，其中，中断为单向输入口，总线阵列相连接的精简高性能总线(AHB)接口以及可选的单周期输入/输出(I/O)口和串行调试或JTAG口为双向口。

32位的Cortex-M0＋处理器核心是计算和控制中心，采用了两级流水的冯·诺依曼结构，执行ARMv6-M指令集(即16位长的Thumb-2指令集，含32位长的扩展指令)，集成了一个单周期的乘法器(用于高性能芯片中)或一个32位的乘法器(用于低功耗芯片中)。

Cortex-M0＋内核具有很强的中断处理能力，一个优先级可配置的嵌套向量中断控制器(NVIC)直接与Cortex-M0＋处理器核心相连接，通过这种紧耦合的嵌套向量中断控制器，可以实现不可屏蔽中断、中断尾链、快速中断响应、睡眠态唤醒中断和四级中断优先

级。这里的“中断尾链”是指当有多个中断被同时触发时，优先级高的中断响应完成后，不需要进行运行环境的恢复，而是直接运行优先级次高的中断，全部中断响应完后，再恢复运行环境。如果没有中断尾链功能，则处理器在响应一个中断前，先进行入栈操作，保存当前中断触发时的运行环境，然后，处理器暂停当前程序的执行去响应中断，响应完中断后，进行出栈操作，恢复响应中断前的环境(即使程序计数器指针(PC)指向被中断的程序位置处)继续执行原来的程序。接着，重复这些操作响应下一个中断。因此，没有中断尾链功能时，两个连续响应的中断中需要插入一次出栈和一次入栈操作(即恢复前一个运行环境和保存后一个运行环境)，而具有中断尾链功能时，这两次堆栈操作均被省略掉了。

Cortex-M0＋内核具有很强的调试能力，通过调试访问口，外部的串行调试或 JTAG 调试口与 Cortex-M0＋处理器的调试器相连接，调试器直接与 Cortex-M0＋处理器核心连接，还通过它与宏跟踪单元相连接。因此，通过串行调试或 JTAG 调试口可以访问 Cortex-M0＋内核的全部资源，包括调试或跟踪程序代码的执行，还可以检查代码的执行结果。

存储保护单元(MPU)可以对存储器的某些空间设定访问权限，使得只有特定的程序代码才能访问这些空间，普通的程序代码则无权访问，这样可以有效地保护关键的程序代码存储区或数据区不受意外访问(例如病毒)的侵害。

图 1－1 中，相对于 Cortex-M0＋处理器核心而言，其余的组件均称为 Cortex-M0＋内核的外设，所有这些组件均为知识产权(IP)核。半导体厂商在这个 IP 核的基础上添加中断发生器、存储器、与 AHB 相连接的多功能芯片外设和输入/输出(I/O)口等，即可以得到特定功能的微控制器芯片。

## 1.3 ARM Cortex-M0＋存储器配置

Cortex-M0＋存储空间的最大访问能力为 $2^{32}$ 字节，即 4 GB。对于 Cortex-M0＋而言，8 位(8 bit)为一个字节，16 位称为半字，32 位称为字。以字为单位，Cortex-M0＋的存储空间的最大访问能力为 $2^{30}$ 字，即从 0 至 $2^{30}-1$ 字。一般地，访问地址习惯采用字节地址，此时，Cortex-M0＋的存储空间配置如图 1－2 所示。

由图 1－2 可知，4 GB 的 Cortex-M0＋映射存储空间被分成 8 个相同大小的空间，每个空间为 0.5 GB。这 8 个空间中，位于地址范围 0x0000 0000～0x1FFF FFFF 的 Code 空间对应着集成在芯片上的 ROM 或 Flash 存储器，主要用于保存可执行的程序代码，也可用于保存数据，其中，中断向量表位于以 0x0 起始的地址空间中，一般占有几十至几百个字节。位于地址范围 0x2000 0000～0x3FFF FFFF 的 SRAM 区域，对应着集成在芯片中的 RAM 存储器，主要用于保存数据，保存的数据在芯片掉电后丢失。位于地址范围 0x4000 0000～0x5FFF FFFF 的片上外设区域，存储着片上外设的访问寄存器，通过读或写这些寄存器，可实现对片上外设的访问和控制。

位于地址范围 0x6000 0000～0x7FFF FFFF 的 RAM 区域属于快速 RAM 区，是具有“写回”特性的缓存区，而位于地址范围 0x8000 0000～0x9FFF FFFF 的 RAM 区域属于慢速 RAM 区，是具有“写通”特性的缓存区。这两个 RAM 区都有对应的缓存(Cache)。所谓的“写回”，是指当 Cortex-M0＋内核向 RAM 区写入数据时，不是直接将数据写入 RAM，而是写入到更快速的缓存中，当缓存满了或者总线空闲时，缓存自动将数据写入到 RAM