ARM

Cortex-M0 LPC1115

开发实战

——芯片级与μC/OS-II系统级

张 勇 吴文华 贾晓天 编著

源程序下载地址：
http://www.buaapress.com.cn的“下载中心”

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

ARM Cortex－M0 LPC1115 开发实战
——芯片级与 μC/OS－II 系统级

张　勇　吴文华　贾晓天　编著

北京航空航天大学出版社

内 容 简 介

本书基于 NXP LPC1115 微控制器和 Keil MDK 集成开发环境，讲述 ARM Cortex－M0 架构、LPC1115 硬件设计以及芯片级别与嵌入式实时操作系统 μC/OS－II 级别的软件设计。全书共分为 15 章，包括 Cortex－M0 内核体系、LPC1115 芯片架构和典型应用电路、Keil MDK 集成开发环境、芯片级 LED 控制、串口通信程序设计以及中断程序设计、μC/OS－II 组件与移植、任务管理与程序框架、系统级中断程序设计、DS18B20 和 LCD 程序设计、I^2C 总线 EEPROM 和 SPI 总线 Flash 访问程序设计、智能密码锁、智能温控报警实例以及 TFT 显示屏驱动程序设计等。本书注重理论与应用的紧密结合，实例丰富，读者可在北京航空航天大学出版社网站下载全部实例工程的源代码。

本书可作为电子通信、软件工程、自动控制、智能仪器和物联网相关专业的高年级本科生或研究生学习微控制器原理和嵌入式操作系统及其应用技术的教材，也可作为嵌入式系统爱好者和开发研究人员的参考用书。

图书在版编目(CIP)数据

ARM Cortex－M0 LPC1115 开发实战：芯片级与 μC/OS－Ⅱ系统级/张勇，吴文华，贾晓天编著. --北京：北京航空航天大学出版社，2014.1

ISBN 978－7－5124－1289－7

Ⅰ. ①A… Ⅱ. ①张… ②吴… ③贾… Ⅲ. ①微处理器－系统设计 Ⅳ. ①TP332

中国版本图书馆 CIP 数据核字(2013)第 254299 号

ARM Cortex－M0 LPC1115 开发实战——芯片级与 μC/OS－II 系统级

张 勇 吴文华 贾晓天 编著

责任编辑 刘 星

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：emsbook@gmail.com 邮购电话：(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本：710×1 000 1/16 印张：23.5 字数：501 千字

2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷 印数：3 000 册

ISBN 978－7－5124－1289－7 定价：49.50 元

前　言

➢ 本书的结构与简介

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，长期以来广泛应用于各种控制和显示等嵌入式系统应用中。随着科技的进步和人们对高智能性设备的喜爱和需求，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 系列传统单片机的新型增强性单片机，例如 TI 公司的 MSP430 系列、Renesas 公司的 RL78 系列、Atmel 公司的 megaAVR 单片机和 Silicon Labs 公司的 C8051F 系列混合型单片机等。新型的单片机具有存储空间大、代码效率高、执行速度快等优点，一定程度上缓解了单片机的应用衰退，但仍然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。

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，当前 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，支持 ARM、Thumb 和 Thumb－2 指令集，主要针对带有操作系统的智能平板电脑；R 系为普通嵌入式内核，支持 ARM、Thumb 和 Thumb－2 指令集，用于嵌入式实时系统；M 系为低功耗系列，仅支持 Thumb－2 指令集，目前有 Cortex－M0、M0＋、M1、M3 和 M4，用于需要快速中断的嵌入式实时应用系统中。Cortex－M3 是最早推出的 Cortex 系列处理器内核，于 2004 年诞生，5 年以后，ARM 公司推出了可商用的 Cortex－M0 内核。Cortex－M 系列的内核中，M0 和 M0＋系列主要针对控制领域，涵盖了传统 8051 系列单片机的应用领域；M3 系列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 系列单片机的应用领域外，还可用于 DSP 处理器的应用领域；M4 系列主要针对高速控制、语音信号处理和数字信号处理领域，包括了传统 8051 系列单片机和 DSP 处理器的应用领域。

NXP(恩智浦)公司是全球最早推出 Cortex－M 系列内核微处理器的公司之一，主要产品有以 LPC1115 微控制器为代表的 LPC11XX 和 LPC12XX 系列(Cortex－M0 内核)、以 LPC812 微控制器为代表的 LPC8XX 系列(Cortex－M0＋内核)、以 LPC1788 微控制器为代表的 LPC13XX、LPC17XX 和 LPC18XX 系列(Cortex－M3

内核)、以 LPC4088 微处理器为代表的 LPC40XX 系列(Cortex - M4 内核)和以 LPC4357 微处理器为代表的 LPC43XX 系列(Cortex - M4 和 M0 双核)。目前,NXP 公司是 Cortex - M 系列微处理器出品最多、型号最全和应用最广的公司之一,NXP 公司的 Cortex - M 系列芯片都体现了低功耗、易使用和高性能的特点,其中,关于 Cortex - M3 内核的 LPC1788 微控制器的应用技术,可参考作者 2013 年在北京航空航天大学出版社出版的《嵌入式实时操作系统 μC/OS - III 应用技术——基于 ARM Cortex - M3 LPC1788》一书。本书将阐述基于 Cortex - M0 核心的 LPC1115 微控制器的系统应用和程序设计方法。

希望上述内容能够回答很多读者关于"为什么学习 ARM Cortex - M0 内核微处理器"和"为什么要学习 LPC1115 微控制器"等问题。为了使本书自成体系,本书包括了 Cortex - M0 体系、LPC1115 架构和芯片级与操作系统级的程序设计等内容,其章节内容概括如下:

第 1 篇(第 1～3 章)为硬件基础,详细介绍了 Cortex - M0 内核体系、LPC1115 微控制器芯片架构和典型应用电路,这部分内容的重点在于 LPC1115 存储配器和 LPC1115 异常与中断向量表。第 3 章所给出的典型应用电路是第 3 篇的程序设计的硬件基础。

第 2 篇(第 4～6 章)为芯片级程序设计,与传统的单片机 C 语言程序设计相似,即使用中断处理外部异步事件,如按键事件等;主程序中使用无限循环(死循环)周期性地显示常规信息。这类程序结构常被称为前台-后台程序结构,前台是无限循环执行的程序,后台是指中断服务程序。在芯片级程序中,直接用 C 语言语句(指针)去访问片上外设,除了外部中断异步事件之外,其他程序代码是按顺序执行的。第 2 篇阐述了 Keil MDK 集成开发环境应用方法和芯片级的 LED 灯、串口和中断程序设计方法。这部分的重点在于 CMSIS 库的使用方法和中断程序设计技巧。

第 3 篇(第 7～15 章)为 μC/OS - II 应用程序设计,先讲述了 μC/OS - II 的工作原理以及在 LPC1115 微控制器上的移植方法;然后介绍了 μC/OS - II 的任务管理方法和基于 μC/OS - II 的程序设计框架;接着,基于 ZLG7289B 芯片阐述了 μC/OS - II 系统中外部中断的响应和处理方法;随后,介绍了单总线温度传感器芯片 DS18B20 的访问方法以及 LCD 屏显示字符和汉字的方法;之后,借助于 EEPROM 芯片 AT24C128 和 Flash 芯片 W25Q64 详细说明了 I^2C 总线和 SPI 总线的使用方法;最后,给出了 2 个有代表性的实例,即智能密码锁和智能温控报警器。为了说明 TFT 型 LCD 屏的显示技术,第 15 章讨论了 LPC1115 驱动 TFT - LCD 屏显示字符和汉字的技术。

➢ 本书的教学思路

本书根据作者的讲义整理扩充而成,理论课时为 48 学时,实验课时为 32 学时,开放实验课时为 16 学时。如果用作大学本科教材,则理论课时宜为 32～48 学时,建

议讲述第1～11章内容，按书中章节顺序讲述；实验学时建议为16～32学时。第13～15章用于课程设计，第12章内容供学生自学讨论。

建议理论教学与实验教学同步进行。理论教学过程中，可以设置2～4学时讨论课，或安排学生分组作学习交流主题报告。实验教学可以设置3～4个基础性实验和1～2个设计性实验，可以结合电子设计大赛的题目开展实验工作。实验应涉及μC/OS-II任务管理以及实验平台外设驱动等方面的内容，应以学生自己动手为主。

➢ 本书的特色

本书具有以下4个方面的特色：

① 详细讲解了基于Cortex-M0核心的LPC1115微控制器存储配置、异常(或中断)向量表以及片上各种外设，结合CMSIS库讲述LPC1115微控制器片上外设的访问方法。

② 详细描述了基于LPC1115的典型应用电路，这些电路涉及LED灯、串口、按键、蜂鸣器、JTAG和ISP电路、复位电路、ADC电路、I^2C总线EEPROM和SPI总线Flash电路、LCD屏电路等。

③ 实例丰富，通过完整的实例详细阐述了芯片级和系统级的程序设计方法，对基于LPC1115微控制器的嵌入式系统软件开发具有较强的指导作用。

④ 对嵌入式实时操作系统μC/OS-II的工作原理、移植以及任务管理等进行了详细的讲解，通过多个实例阐述了μC/OS-II各个组件的使用方法，对学习和应用μC/OS-II具有较好的可借鉴性。

➢ 本书的配套源码

实例丰富是本书的特色，为了节省篇幅，后面章节的工程实例是在前面章节实例的基础上扩充或修改而来的，使得所有的工程实例均为代码完整的实例。因此，读者可以根据本书内容还原书中出现的所有实例工程，作者强烈建议读者按书中介绍的方法和源代码重构各个实例，加深学习认识；此外，读者可通过Email：zhangyong@jxufe.edu.cn向作者索取所有工程源代码，务请来信时告知所使用的仿真器和平台；或者到北京航空航天大学出版社网站下载全部工程源代码。考虑到光盘在运输中易破损，且本书工程程序代码是完整的，故本书没有配套光盘。

➢ 致 射

感谢北京赛佰特公司(http://www.cyb-bot.com)提供了LPC1115实验平台，第4～14章的实例基于赛佰特LPC1115 V1.1平台，第15章的实例基于赛佰特LPC1115 V2.1平台，这两个平台的最大区别在于LPC1115 V1.1开发板使用了基于串口通信的128×64点阵型LCD屏，而LPC1115 V2.1开发板使用了基于并口通信的240×320点阵TFT型LCD屏。需要指出的是，本书内容并不受限于这两个平

台，只要是 LPC1115 平台，甚至是 Cortex－M0 核心的微控制器平台，本书内容都具有较强的针对性。感谢 NXP（恩智浦）公司张宇、辛华峰和王朋朋等领导的大力支持，作者要特别感谢张宇工程师，他多次鼓励作者编写本书并提供了大量建设性意见。还要感谢北京博创兴盛陆海军总经理对本书出版的关心和支持，感谢同事夏家莉、陈滨、蔡鹏、黄坚、张志兵博士以及我的爱人在繁忙的工作之余为我校对书稿。

同时，感谢那些将阅读本书并将提出宝贵意见的读者，这些意见对于作者编写新书或修订再版有实质性的帮助。由于作者水平有限，书中难免会有纰漏之处，敬请专家和读者批评指正。

➢ 免责声明

知识的发展和科技的进步是多元的。本书内容广泛引用了嵌入式实时操作系统 μC/OS－II、Cortex－M0 技术手册、LPC1115 用户手册、Keil MDK 集成开发环境等内容，所有这些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 μC/OS－II、Cortex－M0 核心 LPC1115 微控制器和 Keil MDK 集成开发环境等，禁止任何单位或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。

张 勇

2013 年 10 月

于江西财经大学枫林园

目 录

第 1 篇 硬件基础

第3篇 μC/OS-II应用程序设计

第1篇　硬件基础

本部分内容包括第1～3章，依次为Cortex－M0内核体系、LPC1115芯片架构和LPC1115典型应用电路。这部分内容的重点在于LPC1115存储器配置和异常（或中断）向量表的学习。第3章介绍的LPC1115典型应用电路是后续章节进行软件设计的硬件基础，在后续章节的学习中，要经常对应着本章的电路结构才能更好地理解那些程序代码的工作原理。

第 1 章

Cortex - M0 内核体系

NXP LPC1115 微控制器的内核为 ARM Cortex - M0，本章详细介绍 Cortex - M0 的内核体系。可以这样理解，NXP 公司从 ARM 公司购买了 Cortex - M0 内核知识产权之后，添加上某些特有的外设，然后封装成称为 LPC1115 的芯片。同时，还要理解 Cortex - M0 内核并不是一个实实在在的硬件，而是一些关于内核架构的规定或标准，NXP 公司可以在不违背这些规定和标准的前提下，制做出独具特色的 LPC1115 芯片。例如，不违背 Cortex - M0 内核规定的存储配置方式，设计 LPC1115 芯片的片上存储器大小和映射地址范围。本章内容参考了 ARM 技术手册Cortex - M0 Devices Generic User Guide、Cortex - M0 Technical Reference Manual 和 ARMv6 - M Architecture Reference Manual(均来自 www. arm. com)，建议那些拟深入研究 Cortex - M0 的读者，在学习本章内容的基础上，全面阅读上述文档。

1.1 Cortex - M0 概述

ARM(Advanced RISC Machine)公司推出的 Cortex - M0 内核是一款针对嵌入式控制应用的 32 位微处理器 IP(知识产权)核。ARM 公司本身不生产芯片，仅出售 ARM IP 核，而由其他半导体厂在封装了 ARM IP 核和一些片上外设的基础上，生产各具特色的 ARM 微处理器芯片。Cortex - M0 内核现已经被多家半导体厂商采用，其中最有代表性的是 NXP 公司的 LPC11XX 系列。

Cortex - M0 内核主要针对嵌入式控制领域，相对于传统 8051 单片机，Cortex - M0 内核具有低功耗和执行速度快等优点，低功耗是因为 Cortex - M0 内核门电路少，执行速度快是由于其代码密度高。Cortex - M0 适合用于控制领域是因为其具有先进的确定切换时间的中断控制器。Cortex - M0 IP 核如图 1 - 1 所示。

由图 1 - 1 可知，Cortex - M0 体系由 Cortex - M0 微处理器和可选的两个组件(唤醒中断控制器和调试访问口)组成。Cortex - M0 微处理器包括 Cortex - M0 处理器核心、嵌套向量中断控制器(NVIC)、总线阵列和可选的调试单元(由断点与观测点单元和调试接口组成)。Cortex - M0 微处理器通过精简的 AHB(高性能总线)总线与外部通信。Cortex - M0 微处理器的特点如下：

① ARMv6 - M 体系架构，支持 ARMv6 - M Thumb 指令集，包含 Thumb - 2 指

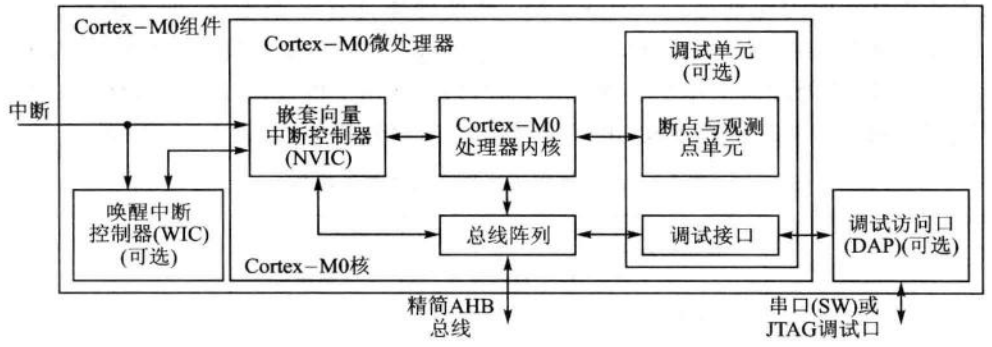

图 1－1　Cortex－M0 IP 核

令集；

② 冯·诺依曼结构，支持 3 级流水线；

③ 具有单周期硬件乘法器或 32 周期硬件乘法器，其中，强调高性能应用对象时提供单周期乘法器，强调小体积应用对象时提供 32 周期乘法器；

④ 32 位微处理器，代码密度高，执行效率高；

⑤ 门电路数目少，功耗低；

⑥ 具有嵌套向量中断控制器，可最多支持 32 个外部中断，4 级中断优先级，低中断切换延时，中断间切换时支持尾链；

⑦ 具有 JTAG 调试口或 2 线的串行 SWD 调试口，其中，SWD 口用于小封装尺寸的 Cortex－M0 微处理器芯片；

⑧ 支持通用的 ARM 仿真开发工具；

⑨ 具有可选的 24 位减计数器 SysTick，可用作实时操作系统时钟节拍。

1.2　Cortex－M0 内核

本节讨论 Cortex－M0 处理器工作模式和内核寄存器等，Cortex－M0 存储模式、异常与中断将在 1.3 节和 1.4 节讲述。

1.2.1　处理器工作模式

Cortex－M0 微处理器具有 2 种工作模式，即线程模式(Thread Mode)和手柄模式(Handler Mode)。Cortex－M0 微处理器芯片复位完成后，进入线程工作模式，线程模式下执行用户应用程序；手柄模式下执行异常(或中断)服务程序，中断返回后进入线程工作模式。

Cortex－M0 支持堆栈由高地址向低地址生长，堆栈指针总是指向最后入栈的数据，当一个新的 32 位数据要入栈时，首先将堆栈指针地址减 4 个字节，然后，数据写

入堆栈指针指向的地址处。Cortex－M0 微处理器具有 2 个堆栈，即主堆栈(Main Stack)和进程堆栈(Process Stack)，具有各自独立的堆栈指针，即 MSP 和 PSP。Cortex－M0 微处理器工作在手柄模式下时，只能使用主堆栈；工作在进程模式下时，可选择使用主堆栈或进程堆栈，由 CONTROL 寄存器控制，如表 1－1 所列。

表 1－1　处理器工作模式与使用的堆栈

使用的堆栈	处理器工作模式	
	线程模式	手柄模式
进程堆栈	√	×
主堆栈	√	√

1.2.2　内核寄存器

Cortex－M0 内核寄存器均为 32 位，包括 13 个通用目的寄存器(R0～R12)、堆栈指针寄存器 SP(R13)、连接寄存器 LR(R14)、程序计数器 PC(R15)、程序状态寄存 PSR、优先级屏蔽寄存器 PRIMASK 和控制寄存器 CONTROL 等，如图 1－2 所示。内核寄存器不是靠地址来访问的，汇编指令中以操作码的形式访问内核寄存器，C 语言往往无法直接访问这些内核寄存器(C 语言指针也无能为力)。

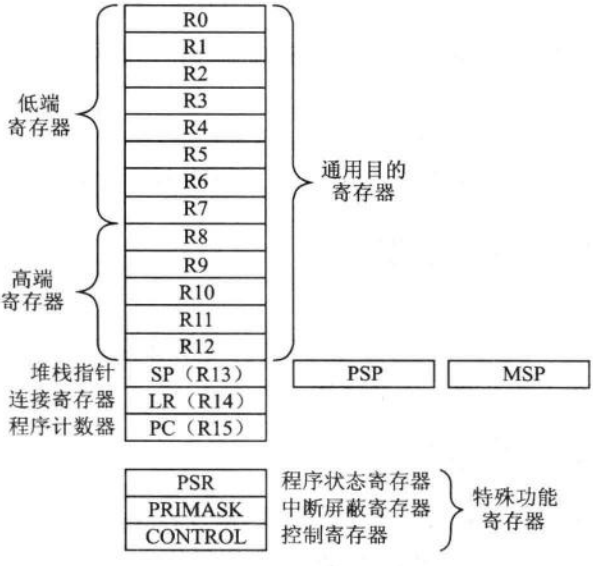

图 1－2　Cortex－M0 内核寄存器

图 1－2 中，SP 寄存器有 2 个映射寄存器。当工作在手柄模式下，SP 寄存器即为 MSP 寄存器；当工作在线程模式下，根据 CONTROL 寄存器将 SP 寄存器设置为 MSP 或 PSP。程序状态寄存器 PSR 包括 3 个寄存器，即应用程序状态寄存器

APSR、中断程序状态寄存器 IPSR 和异常程序状态寄存器 EPSR，其中 IPSR 和 EPSR 为只读寄存器，APSR 为可读/可写寄存器。R0～R15 和 PRIMASK、CONTROL 均为可读/可写寄存器。

下面介绍图 1－2 中各个寄存器的含义。

① R0～R12 为通用目的寄存器，其中，R0～R7 为低端寄存器，可用作所有 16 位或 32 位的指令操作数；R8～R12 为高端寄存器，仅能用作 32 位的指令操作数。

② R13 为堆栈指针 SP，当 CONTROL 寄存器的第 1 位为 0（默认值）时，使用 MSP；当为 1 时，使用 PSP。Cortex－M0 微处理器复位后，地址 0x0 的内容（32 位的数据）装入到 MSP 中，作为主堆栈指针指向的地址。

③ R14 为连接寄存器 LR，当发生子程序跳转、函数调用或异常时，LR 保存程序的返回地址。

④ R15 为程序计数器 PC，包含了当前要执行程序指令的地址，复位时，PC 指向地址 0x04，该地址内容的第 0 位必须为 1，复位时该第 0 位的值 1 写入到 EPSR 寄存器的 T 位，T 位为 1 表示工作在 Thumb 模式下。

⑤ 程序状态寄存器 PSR 包括 3 个寄存器，即 APSR、IPSR 和 EPSR，这些寄存器均为 32 位寄存器，但是绝大多数位均为保留的位，其结构如图 1－3 所示。

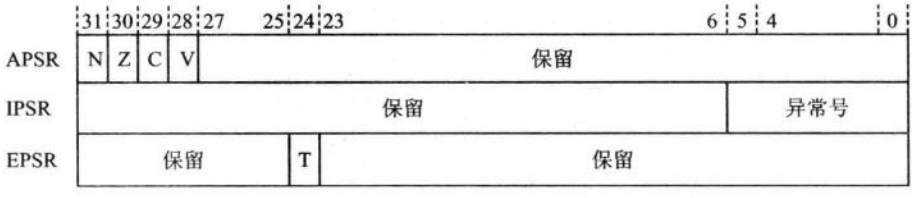

图 1－3　程序状态寄存器

图 1－3 中，IPSR 和 EPSR 为只读存储器，APSR 为可读/可写存储器，这 3 个寄存器可单独访问，也可以组合访问。组合访问情况下，PSR 表示 APSR＋EPSR＋IPSR，IEPSR 表示 EPSR＋IPSR，IAPSR 表示 APSR＋IPSR，EAPSR 表示 APSR＋EPSR。访问指令为：读指令为 MRS，写指令为 MSR。PSR 各位的含义为：N、Z、C 和 V 为标志位，分别表示运算结果的负标志、零标志、进位或借位标志和溢出标志。IPSR[5:0]用于保存异常号，即当某个异常或中断发生后，这个异常或中断的异常号或中断号将被保存在 IPSR[5:0]中，其值含义为：0 表示线程模式，1、4～10、12、13 均为保留，2 表示不可屏蔽中断发生了，3 表示硬件访问出错（HardFault）异常发生了，11 表示超级用户特权调用（SVCall）异常发生了，14 表示请求特权服务（PendSV）异常发生了，15 表示系统时钟节拍中断发生了，n＋16（31≥n≥0）表示第 n 号中断（IRQn）发生了。EPSR 寄存器的 T 位表示 Thumb 模式位，必须为 1。

⑥ 中断屏蔽寄存器 PRIMASK 只有第 0 位有效，其余各位均保留。第 0 位为 1，关闭所有可屏蔽异常和中断；为 0，表示开放这些异常和中断。

⑦ 控制寄存器 CONTROL 只有第 1 位有效，其余位均保留。第 1 位为 0，表示 MSP 为当前程序使用的堆栈；为 1，表示 PSP 为当前堆栈。在手柄模式下，由于只能使用 MSP，此时，该位读出为 0，写入无效。

1.3　Cortex－M0 存储配置

Cortex－M0 微处理器最大支持 4 GB 的映射存储空间，如图 1－4 所示。

图 1－4　Cortex－M0 映射存储空间

根据图 1－4，地址空间 0x0～0x1FFF FFFF 为代码区，用于存放可执行程序代码，也可以存放数据；地址空间 0x2000 0000～0x3FFF FFFF 为 SRAM 区，用于存放可执行程序的数据，也可以存放代码；地址空间 0x4000 0000～0x5FFF FFFF 为片上外设区，用于保存外设寄存器；0x6000 0000～0x9FFF FFFF 为外部 RAM 区，用于存放数据或程序代码；地址空间 0xA000 0000～0xDFFF FFFF 为外部设备区；地址空间 0xE000 0000～0xE00F FFFF 为私有外设总线区域，包括 NVIC、系统定时器和系统控制模块等，仅可按字访问；地址空间 0xE010 0000～0xFFFF FFFF 为专用设备存储区。

Cortex－M0 为 32 位微处理器，其字数据存储方式有 2 种，即小端模式和字节保序大端模式。一个字数据由 4 个字节组成，当字数据保存在存储空间时，低字节保存在低地址，高字节保存在高地址，称为小端模式；如果低字节保存在高地址，而高字节保存在低地址，并且每个字节内的位顺序不变，称为字节保序大端模式，如图 1－5 所示。