

 高等教育规划教材

基于ARM Cortex-M4的单片机 原理与实践

陈朋 梁荣华 刘义鹏 等编著



免费提供电子教案、习题答案、源代码

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

高等教育规划教材

基于 ARM Cortex-M4 的单片机 原理与实践

陈 朋 梁荣华 刘义鹏 等编著



机械工业出版社

本书以德州仪器公司的 TM4C1294 微处理器为蓝本,由浅入深地介绍了 ARM Cortex-M4 内部结构、特点及片上功能模块的工作原理、操作方法以及集成开发环境 (Code Composer Studio, CCS) 的使用方法。本书还阐述了 TM4C1294 微处理器系统外设、串行通信外设接口、模拟外设以及运动控制外设的功能特点、内部结构、初始化与配置以及寄存器映射与描述等。本书在最后介绍了基于 TM4C12x 和 TM4C1294 微处理器的综合应用实例,可使读者迅速掌握基于 ARM Cortex-M4 微处理器的应用技巧,并可向读者提供配套开发板。

本书既可以作为电子信息工程、自动化、电子科学与技术、通信工程、光电信息科学与工程、信息工程及相关专业的本专科生或研究生学习微处理器原理的教材,也可以作为相关专业技术人员的参考用书。本书注重知识点与实践相结合。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取 (QQ: 2850823885。电话: 010-88379739)。

图书在版编目 (CIP) 数据

基于 ARM Cortex-M4 的单片机原理与实践/陈朋等编著. —北京: 机械工业出版社, 2018. 9

高等教育规划教材

ISBN 978-7-111-60347-4

I. ①基… II. ①陈… III. ①单片微型计算机-高等学校-教材
IV. ①TP368. 1

中国版本图书馆 CIP 数据核字 (2018) 第 150199 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 胡 静 责任校对: 张艳霞

责任印制: 张 博

三河市宏达印刷有限公司印刷

2018 年 6 月第 1 版·第 1 次印刷

184mm×260mm·18.25 印张·446 千字

0001-3000 册

标准书号: ISBN 978-7-111-60347-4

定价: 55.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

服务咨询热线: (010) 88379833

读者购书热线: (010) 88379649

封面防伪标均为盗版

网络服务

机工官网: www.cmpbook.com

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

金书网: www.golden-book.com

前 言

当今, ARM 技术被广泛应用, 领域涉及手机、数字机顶盒以及汽车制动系统和网络路由器等, 并且迅速向传统的嵌入式领域渗透。全球 95% 以上的手机以及超过四分之一的电子设备都在使用 ARM 技术。

近些年来, 随着电子制造工艺的不断发展进步, ARM Cortex-M4 微处理器的成本也在不断降低, 已经与 8 位和 16 位微处理器的成本处于同等水平。如今, 越来越多的微处理器供应商提供基于 ARM 的微处理器, 这些产品在外设、性能、内存大小、封装以及成本等方面具有越来越多的选择。其中, 基于 ARM Cortex-M4 微处理器是由 ARM 专门开发的新型嵌入式处理器, 用以满足有效且易于使用的控制和信号处理功能混合的数字信号控制市场。

本书基于 ARM Cortex-M4 内核的 TM4C1294 微处理器和 CCS (Code Composer Studio) 集成开发环境详细介绍了 ARM Cortex-M4 原理与实践。

本书注重知识点与实践相结合。在概述 ARM Cortex-M4 微处理器资源后, 介绍了 Cortex-M4F 微处理器的系统控制内容。之后介绍了 CCS 集成开发环境的使用, 包括 CCS 工程的建立以及调试等。从第 4 章开始, 在每一章节都会配有几个应用例程, 供读者参考。所配的程序有的来自 TI 官方网站, 作为基础巩固; 有的来自编者项目开发中的实例, 作为拓展提高。读者在每学习完一个外设模块后, 都可以结合本书的应用例程, 编写对应的应用程序, 并在 Cortex-M4 开发板上调试, 进而让读者在实际操作中掌握该外设模块的操作方法, 真正实现理论与实践相结合。本书所附程序均已在 Cortex-M4 开发板上调试通过。

本书共 9 章, 具体内容如下。

第 1 章绪论, 从总体上介绍 ARM 体系, 以及 Cortex 内核的分类及特点, 然后重点阐述 ARM Cortex-M4 微处理器的特点以及 TI 公司的 Cortex-M4 芯片 TM4C1294。

第 2 章介绍 Cortex-M4F 微处理器的系统控制, 包括系统控制相关信号的描述以及系统控制的功能概述。

第 3 章介绍 TI 公司集成开发环境 CCS 的基本知识及操作。

第 4 章介绍 TM4C1294 微处理器的内部存储器和外部扩展接口, 包括整体功能框图, SRAM、ROM、Flash Memory 和 EEPROM 的功能描述和相关寄存器映射, 最后介绍了外部总线扩展接口 EPI。

第 5 章重点阐述 TM4C1294 微处理器的系统外设, 包括通用输入/输出端口 (GPIO), 通用定时器模块 (GPTM), 看门狗定时器 (WDT) 和微型直接存储器访问 (μ DMA) 等模块的功能特点、内部结构、初始化与配置以及寄存器映射与描述。

第 6 章介绍 TM4C1294 微处理器的串行通信外设接口, 包括通用异步接收/发送装置 UART、同步串行接口 SSI、I²C 总线、CAN 总线、USB 总线和以太网控制器等模块的功能特点、内部结构、初始化与配置以及寄存器映射与描述。

第 7 章介绍 TM4C1294 微处理器的模拟外设, 包括模拟比较器和模-数转换器 (ADC) 模块的功能特点、内部结构、初始化与配置以及寄存器映射与描述。

第 8 章介绍 TM4C1294 微处理器的运动控制外设,包括脉冲宽带调制 (PWM) 和正交编码接口 (QEI) 模块的功能特点、内部结构、初始化与配置以及寄存器映射与描述。

第 9 章介绍基于 TM4C12x 的应用实例,包括基于 TM4C123 Launchpad 的步进电动机驱动应用设计以及 Wi-Fi 应用,基于 TM4C1294 的加速度计重力感应游戏、音乐播放器设计以及贪吃蛇游戏。

参与本书编写工作的主要人员有陈朋、梁荣华和刘义鹏,最终方案的确定和本书的定稿工作由陈朋负责。德州仪器半导体技术(上海)有限公司王沁工程师负责第 9 章前 3 节内容的编写。浙江工业大学梅一珉、欧林林、邢科新、陈德富和禹鑫焱教师参加了本教材的试用,并提供了具体的修改意见。感谢浙江工业大学控制科学与工程学科对本书出版的支持。感谢机械工业出版社的编辑,他们在本书的创作与出版过程中提出了很多宝贵意见,使本书不断完善。

由于作者水平和实践能力有限,书中难免会存在不足和疏漏之处,恳请读者提出宝贵的意见,以便以后予以补充和修订。

本书在出版之前,已经作为讲义在编者学校本科学生中试用 3 年,融合编者多年工作经验和心得体会。本书的反馈邮箱为 chenpeng@zjut.edu.cn,真诚希望得到来自读者的宝贵意见和建议。

编者

目 录

前言

第1章 绪论	1
1.1 ARM体系概述	1
1.1.1 ARM简介	1
1.1.2 ARM架构的发展	2
1.1.3 Cortex内核分类	3
1.2 ARM Cortex-M4微处理器	4
1.2.1 ARM Cortex-M4微处理器特点	4
1.2.2 Cortex-M4微处理器结构	5
1.2.3 Cortex-M4微处理器的编程模式	8
1.2.4 Cortex-M4微处理器的堆栈	8
1.2.5 Cortex-M4微处理器的内核寄存器	8
1.2.6 Cortex-M4微处理器的存储器映射	9
1.2.7 Cortex-M4指令集	10
1.2.8 Cortex-M4的中断与异常处理	11
1.3 TM4C1294微处理器	13
1.3.1 TM4C1294微处理器概述	13
1.3.2 TM4C1294微处理器结构	13
1.3.3 TM4C1294微处理器性能特点	15
1.4 思考与练习	16
第2章 Cortex-M4F微处理器的系统控制模块	17
2.1 系统控制模块的相关信号描述	17
2.2 系统控制模块的功能概述	17
2.2.1 器件标识信息	17
2.2.2 复位控制	18
2.2.3 NMI控制	22
2.2.4 电源控制	23
2.2.5 时钟控制	23
2.2.6 工作模式控制	27
2.2.7 系统初始化与配置	28
2.3 思考与练习	29
第3章 CCS集成开发环境	30

3.1 集成开发环境 CCS	30
3.1.1 CCS功能及特点	30
3.1.2 安装 CCSv6	31
3.1.3 启动 CCSv6	32
3.1.4 新建 CCS工程	34
3.1.5 建立工程	41
3.1.6 基本调试功能	42
3.1.7 使用观察窗口	43
3.1.8 CCS的其他基本操作	44
3.1.9 CCS编程简介	45
3.2 TivaWare软件	47
3.2.1 TivaWare功能及特点	47
3.2.2 TivaWare主要模块介绍	47
3.3 思考与练习	49
第4章 TM4C1294微处理器内部存储器和外部扩展接口	50
4.1 TM4C1294片内存储器功能框图	50
4.2 TM4C1294片内存储器功能描述	51
4.2.1 SRAM	51
4.2.2 ROM	52
4.2.3 Flash Memory	54
4.2.4 EEPROM	60
4.3 TM4C1294寄存器映射与描述	62
4.4 TM4C1294外部总线扩展接口(EPI)	64
4.4.1 EPI功能与特点	64
4.4.2 EPI内部结构	65
4.4.3 EPI功能描述	66
4.4.4 EPI初始化与配置	68
4.4.5 EPI寄存器映射	77
4.4.6 EPI应用例程	78
4.5 思考与练习	85
第5章 TM4C1294微处理器系统外设	86

5.1 通用输入/输出端口 (GPIO) ...	86	6.2.2 QSSI 内部结构	136
5.1.1 GPIO 功能与特点	86	6.2.3 QSSI 功能描述	136
5.1.2 GPIO 功能描述	87	6.2.4 QSSI 初始化与配置	139
5.1.3 GPIO 初始化与配置	91	6.2.5 QSSI 寄存器映射与描述	141
5.1.4 GPIO 寄存器映射与描述	92	6.2.6 QSSI 应用例程	142
5.1.5 GPIO 应用例程	94	6.3 I ² C 总线	142
5.2 通用定时器模块 (GPTM)	95	6.3.1 I ² C 功能与特点	143
5.2.1 GPTM 功能与特点	95	6.3.2 I ² C 内部结构	143
5.2.2 GPTM 内部结构	96	6.3.3 I ² C 功能描述	144
5.2.3 GPTM 功能描述	96	6.3.4 I ² C 初始化与配置	148
5.2.4 GPTM 初始化与配置	104	6.3.5 I ² C 寄存器映射与描述	150
5.2.5 GPTM 寄存器映射与描述	106	6.3.6 I ² C 应用例程	151
5.2.6 GPTM 应用例程	107	6.4 CAN 总线	152
5.3 看门狗定时器 (WDT)	108	6.4.1 CAN 功能与特点	152
5.3.1 WDT 功能与特点	109	6.4.2 CAN 控制器内部结构	152
5.3.2 WDT 内部结构	109	6.4.3 CAN 功能描述	153
5.3.3 WDT 功能描述	109	6.4.4 CAN 初始化与配置	159
5.3.4 WDT 初始化与配置	110	6.4.5 CAN 寄存器映射与描述	160
5.3.5 WDT 寄存器映射与描述	111	6.4.6 CAN 应用例程	161
5.3.6 WDT 应用例程	111	6.5 通用串行总线 (USB)	162
5.4 微型直接存储器访问 (μ DMA) ...	112	6.5.1 USB 功能与特点	163
5.4.1 μ DMA 控制器功能与特点	112	6.5.2 USB 内部结构	163
5.4.2 μ DMA 控制器内部结构	113	6.5.3 USB 功能描述	164
5.4.3 μ DMA 控制器功能描述	113	6.5.4 USB 初始化与配置	171
5.4.4 μ DMA 控制器初始化与配置	121	6.5.5 USB 寄存器映射与描述	173
5.4.5 μ DMA 通道控制结构体	125	6.6 以太网控制器	179
5.4.6 μ DMA 寄存器映射与描述	125	6.6.1 以太网控制器的功能与特点	180
5.4.7 μ DMA 应用例程	126	6.6.2 以太网控制器的内部结构	180
5.5 思考与练习	127	6.6.3 以太网控制器的功能描述	181
第 6 章 TM4C1294 微处理器的串行		6.6.4 以太网控制器的初始化与配置 ...	182
通信外设接口	128	6.6.5 以太网控制器的寄存器映射与	
6.1 通用异步收发器 (UART)	128	描述	183
6.1.1 UART 功能与特点	128	6.7 思考与练习	186
6.1.2 UART 内部结构	129	第 7 章 TM4C1294 微处理器的模拟	
6.1.3 UART 功能描述	130	外设	188
6.1.4 UART 初始化与配置	133	7.1 TM4C1294 微处理器的模拟	
6.1.5 UART 寄存器映射与描述	134	比较器	188
6.1.6 UART 应用例程	135	7.1.1 模拟比较器的内部结构	188
6.2 四路同步串行接口 (QSSI) ...	135	7.1.2 模拟比较器的功能描述	188
6.2.1 QSSI 功能与特点	136	7.1.3 模拟比较器的内部参考电压	
		编程	188

7.1.4 模拟比较器的初始化与配置	190	9.1.3 软件介绍	228
7.1.5 模拟比较器的寄存器映射与描述	190	9.2 基于 TM4C123 LaunchPad 的步进电动机驱动应用设计	232
7.1.6 模拟比较器的应用例程	191	9.2.1 TM4C123GH6PM 微处理器介绍	232
7.2 TM4C1294 微处理器的模-数转换器 (ADC)	191	9.2.2 DRV8833 步进电动机驱动器	232
7.2.1 ADC 功能与特点	192	9.2.3 系统硬件	235
7.2.2 ADC 内部结构	192	9.2.4 系统软件	236
7.2.3 ADC 功能描述	194	9.3 基于 TM4C1294 LaunchPad 的 Wi-Fi 应用	240
7.2.4 ADC 初始化与配置	198	9.3.1 TM4C1294 和 CC3100 介绍	240
7.2.5 ADC 寄存器映射与描述	199	9.3.2 系统硬件	242
7.2.6 ADC 的应用例程	201	9.3.3 系统软件	244
7.3 思考与练习	203	9.4 基于 TM4C1294 的 AY-SCMP Kit 实验开发板硬件系统介绍	250
第 8 章 TM4C1294 微处理器的运动控制外设	204	9.4.1 系统组成和功能框图	250
8.1 脉冲宽度调制 (PWM)	204	9.4.2 系统部分硬件资源	250
8.1.1 PWM 功能与特点	204	9.5 基于 TM4C1294 和加速度计的重力感应游戏	267
8.1.2 PWM 内部结构	205	9.5.1 重力感应游戏概述	267
8.1.3 PWM 功能描述	206	9.5.2 系统软件	268
8.1.4 PWM 初始化与配置	209	9.5.3 实验结果展示	270
8.1.5 PWM 寄存器映射	210	9.6 基于 TM4C1294 的音乐播放器设计	271
8.1.6 PWM 应用例程	212	9.6.1 音乐播放器设计思路概述	271
8.2 正交编码器接口 (QEI)	215	9.6.2 工作原理	272
8.2.1 QEI 功能与特点	215	9.6.3 软件设计	272
8.2.2 QEI 内部结构	215	9.6.4 实验结果展示	274
8.2.3 QEI 功能描述	216	9.7 基于 TM4C1294 贪吃蛇游戏设计	275
8.2.4 QEI 初始化与配置	218	9.7.1 贪吃蛇游戏设计概述	275
8.2.5 QEI 寄存器映射与描述	219	9.7.2 系统软件	275
8.2.6 QEI 应用例程	219	9.7.3 实验结果展示	279
8.3 思考与练习	220	附录	280
第 9 章 基于 TM4C12x 的综合应用实例	221	附录 A TM4C1294 引脚定义	280
9.1 基于 TM4C123 LaunchPad 的硬件平台介绍	221	参考文献	284
9.1.1 硬件平台性能概述	222		
9.1.2 硬件平台功能模块介绍	222		

第1章 绪论

当今，ARM 技术被广泛应用，领域涉及手机、数字机顶盒以及汽车制动系统和网络路由器等，并且迅速向传统的嵌入式领域渗透。全球 95% 以上的手机以及超过四分之一的电子设备都在使用 ARM 技术。

ARM Cortex-M4 微处理器是由 ARM 专门开发的新型嵌入式处理器，用以满足有效且易于使用的控制与信号处理功能混合的数字信号控制市场。

1.1 ARM 体系概述

ARM 是 Advanced RISC Machines 的缩写，有 3 个含义。

- 一个生产高级 RISC（精简指令集）微处理器的公司。
- 一种高级 RISC 的技术。
- 一类采用高级 RISC 的微处理器。

1.1.1 ARM 简介

1. ARM 公司

ARM 公司成立于 1990 年，总部位于英国剑桥，是全球领先的半导体知识产权（IP）提供商之一，它拥有 1700 多名员工，在全球设立了多个办事处，其中包括比利时、法国、印度、瑞典和美国的设计中心。

ARM 公司的商业模式主要是 IP 的设计和许可，它本身并不生产和销售实际的半导体芯片，而是向对这些 IP 设计有兴趣的公司和厂商授予 IP 许可证，同时收取 IP 的许可费用和生产芯片、晶片的版税。

对于每一个授权公司和厂商来说，它们获得的授权都是独一无二的。它们可以根据不同的应用领域和自身的技术优势，适当加入外围电路，形成自己的 ARM 微处理器芯片，从而缩短了开发周期，提升了产品的竞争力。

到目前为止，ARM 公司向 200 多家公司出售了 600 多个处理器许可证，全球已经累计销售了超过 150 亿块基于 ARM 的芯片。

2. RISC（精简指令集）

RISC（Reduced Instruction Set Computing），即精简指令集。RISC 与 CISC（Complex Instruction Set Computing，复杂指令集）相对应，CISC 采用的优化方法是通过设置一些功能复杂的指令，把一些原来由软件实现的、常用的功能改用硬件的指令系统实现，以此来提高计算机的执行速度；RISC 技术的精华就是通过简化指令功能，使指令的平均执行周期缩短，从而提高工作主频，同时大量使用通用寄存器来提高子程序执行的速度。RISC 与 CISC 是指令系统两个截然不同的优化方向。

RISC 的基本思想为尽量简化指令功能，只保留那些功能简单、能在一个节拍内执行完

成的指令，而把较复杂的功能用一段子程序来实现。

3. ARM 处理器

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，其设计的 ARM 系列处理器均采用了精简指令集的设计。如今内嵌有 ARM 处理器核心和采用 ARM 构架的处理器，都被统称为 ARM 处理器。

1991 年，ARM 公司推出第一款嵌入式 RISC 核心，即 ARM6 解决方案。1993 年，ARM 公司推出基于 ARMv4T 架构的 ARM7 核心。2004 年，ARM 公司发布基于 ARMv7 架构的 Cortex 微处理器系列，同时发布该系列架构中首款 ARM Cortex-M3 微处理器。2010 年，ARM 公司推出了 ARM Cortex-M4 微处理器。

Cortex-M4 是一种面向数字信号处理（DSC）和高级微控制器（MCU）应用的高效方案，具有高效率的信号处理能力，同时还有低功耗、低成本和简单易用等特点，适合电机控制、汽车、电源管理、嵌入式音频和工业自动化等领域。

Cortex-M4 微处理器内集成了单循环乘法累计（MAC）单元、优化的单指令多数据（SIMD）指令、饱和算法指令和可选择的单精度浮点单元（FPU），同时保留了 Cortex-M 系列的一贯特色技术，例如，处理性能最高 1.25DMIPS/MHz 的 32 位核心、代码密度优化的 Thumb-2 指令集、负责中断处理的嵌套中断向量控制器。此外，还可以选择内存保护单元（MPU）、低成本诊断和追踪以及完整休眠状态。

Cortex-M4 可以根据应用需要提供多种不同的制造方式，例如，超低功耗版本采用台积电 180 nm ULL 工艺生产，目标频率 150 MHz 的高性能版本则使用 GLOBALFOUNDRIES 公司 65 nm LPe 工艺生产，动态功耗也不超过 40 μ W/MHz。

现在已经有 5 家 MCU 半导体企业购买了 ARM Cortex-M4 的授权，包括 NXP、意法半导体和德州仪器等行业巨头。

1.1.2 ARM 架构的发展

ARM 处理器架构简单，使用较小的集成电路规模，从而降低了功耗，同时实现了较高的性能和代码密度，达到了良好的平衡，这是 ARM 架构的关键特性。

ARM 架构一方面在各个版本之间保持了很好的兼容性，另一方面也在不断地改进。架构进化历史如图 1-1 所示。

ARM 处理器的架构特点如下。

1) ARMv4T 架构。引进了 16 位 Thumb 指令集和 32 位 ARM 指令集，目的是在同一个架构中同时提高性能和代码密度。16 位 Thumb 指令集相对于 32 位 ARM 指令集可缩减高达 35% 的代码大小，同时保持 32 位架构的优点。采用此架构的内核有 ARM7TDMI，具体芯片有三星 S3C44B0x 系列等。

2) ARMv5TEJ 架构。引进了数字信号处理（Digital Signal Processing, DSP）算法（如饱和运算）和 Jazelle Java 字节码引擎来启用 Java 的硬件执行，从而提高了用 Java 编写的应用程序的性能。与非 Java 加速内核比较，Jazelle 将 Java 的执行速度提高了 8 倍，并且减少了 80% 的功耗。许多基于 ARM 处理器的便携式设备中已使用此架构，目的是在游戏和多媒体应用程序的性能方面提供有显著改进的用户体验。采用此架构的内核有 ARM926EJ-S，具体芯片有 ATMEL 的 AT91SAM926x 系列等。

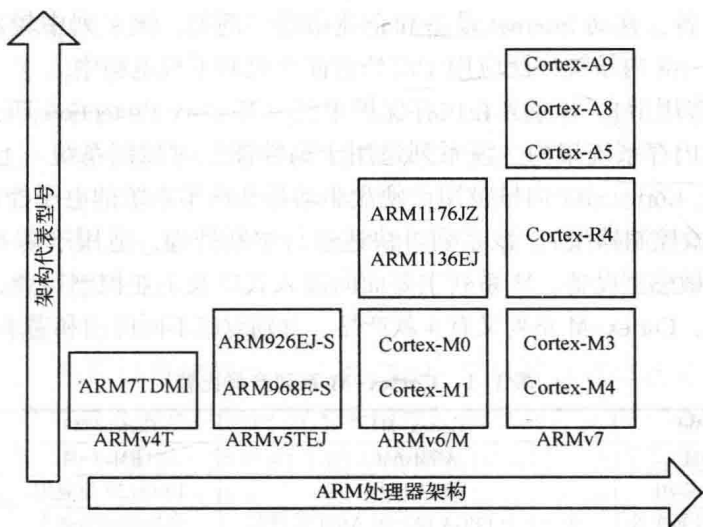


图 1-1 ARM 处理器架构进化历史

3) ARMv6 架构。引进了包括单指令多数据 (Single Instruction Multiple Data, SIMD) 运算在内的一系列新功能。SIMD 扩展已针对多种软件应用程序 (包括视频编解码器和音频编解码器) 进行优化, 对于这些软件应用程序, SIMD 扩展最多可将性能提升 4 倍。此外, 还引进了作为 ARMv6 架构变种的 Thumb-2 和 TrustZone 技术。采用此架构的内核有 ARM1176JZ, 具体芯片有三星 S3C6410x 系列等。

4) ARMv6M 架构。为低成本、高性能设备而设计, 向由 8 位设备占主导地位的市场提供 32 位功能强大的解决方案。其 16 位 Thumb 指令集架构允许设计者设计门数最少却十分经济实惠的设备。采用此架构的内核有 Cortex-M0, 具体芯片有 ST 的 STM32F0 系列等。

5) ARMv7 架构。是目前 ARM 公司最新的架构, 所有 Cortex 微处理器都实现了 ARMv7 架构 (ARMv6M 的 Cortex-M 系列微处理器除外)。所有 ARMv7 架构都实现了 Thumb-2 技术 (一个经过优化的 16/32 位混合指令集), 在保持与现有 ARM 解决方案的代码完全兼容的同时, 既具有 32 位 ARM 指令集的性能优势, 又具有 16 位 Thumb 指令集的代码大小优势。ARMv7 架构还包括 NEON 媒体加速技术, 该技术可将 DSP 和媒体吞吐量提升 400%, 并提供改进的浮点支持以满足下一代 3D 图形和游戏物理学以及传统嵌入式控制应用程序的需要。

早期 ARM 处理器使用一种基于数字的命名方法, 数字之后添加字母后缀, 但该数字并不表示其内核架构, 如 ARM7TDMI 并不是一款 ARMv7 内核微处理器, 而是 ARMv4T 架构的产品。

1.1.3 Cortex 内核分类

Cortex 系列内核的命名采用 Cortex 加后缀的方式。后缀用字母加数字的方式表示其产品特性, 如 M3。Cortex 分为 3 个系列。

1) A 系列 (应用程序型)。A 系列在内存管理单元 (Memory Management Unit, MMU)、用于多媒体应用程序的可选 NEON (加速多媒体和信号处理算法) 处理单元以及支持半精度和精度运算的高级硬件浮点单元的基础上实现了虚拟内存系统架构。该系列适用于高端消费

电子设备、网络设备、移动 Internet 设备和企业市场。例如，该系列中较新的 Cortex-A9 内核和之前的 Cortex-A8 内核被广泛应用于高档智能手机和平板电脑中。

2) R 系列（实用型）。R 系列在内存保护单元（Memory Protection Unit, MPU）的基础上实现了受保护的内存系统架构。该系列适用于高性能实时控制系统（包括汽车和大容量存储设备）。例如，Cortex-R4 内核被用于硬盘驱动器和汽车系统的电子控制单元中。

3) M 系列（微控制器型）。该系列可快速进行中断处理，适用于需要高度确定的行为和门数最少的成本敏感型设备。M 系列主要面向嵌入式以及工业控制行业，用来取代“旧时代”单片机。其中，Cortex-M 系列又有 4 款产品，分别对应不同应用和需求，如表 1-1 所示。

表 1-1 Cortex-M 系列产品比较

名称	Cortex-M0	Cortex-M1	Cortex-M3	Cortex-M4
架构	ARMv6M	ARMv6M	ARMv7-M	ARMv7E-M
应用范围	8 位/16 位应用	FPGA 应用	16 位/32 位应用	32 位/DSC 应用
特点	成本低，具有简单性	第一个为 FPGA 设计的 ARM 处理器	高性能和高效率	有效的数字信号控制

1.2 ARM Cortex-M4 微处理器

ARM Cortex-M4 微处理器高效的信号处理功能与 Cortex-M 微处理器系列的低功耗、低成本和易于使用的优点组合，旨在满足专门面向电动机控制、汽车、电源管理、嵌入式音频和工业自动化市场的新兴类别的灵活解决方案。

1.2.1 ARM Cortex-M4 微处理器特点

1. 高效数字信号控制

Cortex-M4 提供了无可比拟的功能，以将 32 位控制与领先的数字信号处理技术集成来满足需要很高能效级别的市场。Cortex-M4 微处理器采用一个扩展的单时钟周期乘法累加（Multiply and Accumulate, MAC）单元、优化的 SIMD 指令、饱和运算指令和一个可选的单精度浮点单元（Float Point Unit, FPU）。这些功能以表现 ARM Cortex-M 系列微处理器特征的创新技术为基础。

1) RISC 微处理器内核。高性能 32 位 CPU、具有确定性的运算、低延迟 3 阶段管道，可达 1.25DMIPS/MHz。

2) Thumb-2 指令集。16 位/32 位指令的最佳混合、小于 8 位设备 3 倍的代码大小、对性能没有负面影响，提供最佳的代码密度。

3) 低功耗模式。集成的睡眠状态支持、多电源域和基于架构的软件控制，可满足低功耗要求。

4) 嵌套向量中断控制器（Nested Vectored Interrupt Controller, NVIC）。低延迟、低抖动中断响应、不需要汇编编程和以纯 C 语言编写的中断服务例程，能出色完成中断处理。

5) 工具和实时操作系统（Real Time Operating System, RTOS）支持。广泛的第三方工具支持、Cortex 微控制器软件接口标准（Cortex Microcontroller Software Interface Standard, CMSIS）、最大限度地增加软件成果重用。实时操作系统是指当外界事件或数据产生时，能够接受并以足够快的速度予以处理，其处理的结果又能在规定的时间内来控制生产过程或

对处理系统作出快速响应，并控制所有实时任务协调一致运行的操作系统。因而，提供及时响应和高可靠性是其主要特点。

6) CoreSight 调试和跟踪。JTAG (Joint Test Action Group) 接口或 2 针串行线调试 (Serial Wire Debug, SWD) 连接，支持多处理器和实时跟踪。

此外，该微处理器还提供了一个可选的 MPU，提供低成本的调试、追踪功能和集成的休眠状态，以增加灵活性。嵌入式开发者可以快速设计并推出令人瞩目的终端产品，具备最多的功能以及最低的功耗和尺寸。

2. 易于使用的技术

Cortex-M4 通过一系列出色的软件工具和 CMSIS 使信号处理算法开发变得十分容易。CMSIS 是 Cortex-M 系列微处理器与供应商无关的硬件抽象层。使用 CMSIS，可以为接口外设、实时操作系统和中间件提供一致且简单的处理器软件接口，从而简化软件的重用。使用 CMSIS 可缩短新微控制器开发人员的学习过程，从而缩短新产品的上市时间。

ARM 目前正在对 CMSIS 进行扩展，将加入支持 Cortex-M4 扩展指令集的 C 编译器；同时，ARM 也在开发一个优化库，方便 MCU 用户开发信号处理程序。该优化库将包含数字滤波算法和其他基本功能，例如，数学计算、三角计算和控制功能。数字滤波算法也将可以与滤波器设计工具和设计工具包（如 MATLAB 和 LabVIEW）配套使用。

1.2.2 Cortex-M4 微处理器结构

Cortex-M4 微处理器的结构如图 1-2 所示，其中包括处理器内核 CM4 Core、内核外设、调试和跟踪接口以及多条总线接口。

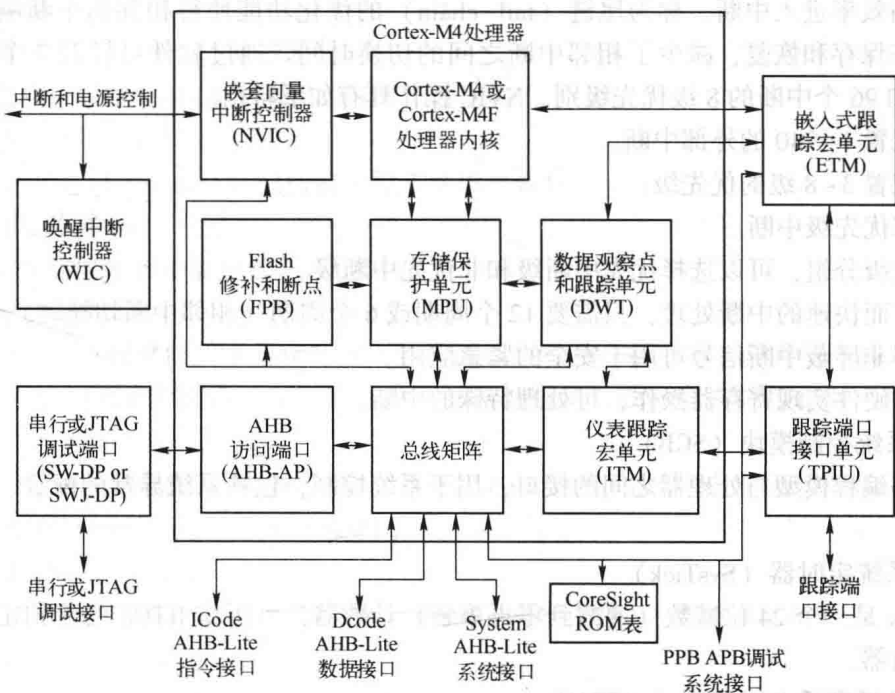


图 1-2 Cortex-M4 微处理器的结构

1. 内核外设

内核外设包括存储保护单元 (MPU)、嵌套向量中断控制器 (NVIC)、系统控制模块 (System Control Block, SCB) 和系统定时器 (System Timer, SysTick)。另外, 在 Cortex-M4F 中, 还有单精度浮点运算单元 (FPU)。

(1) 存储保护单元 (MPU)

MPU 是 Cortex-M4 中用于内存保护的可选组件。MPU 支持标准的 ARMv7 保护存储系统结构 PMSA 模型。MPU 将存储器分为若干区域, 定义各区域的位置、大小、访问权限和存储属性等。MPU 可为每一个区以及重叠区单独设置存储属性, 还可将存储属性导出至系统。

存储属性确定存储区的访问行为。Cortex-M4 微控制器定义了 8 个独立的存储区域即 0~7 区域和一个背景区域。背景区域的存储属性即为默认的存储器映射的属性, 可以在特权软件执行模式下访问。当存储区域发生重叠时, 存储访问受存储属性最高的区域的影响。例如, 区域 7 的存储属性将被任何与其重叠的区域优先采用。

如果程序访问被 MPU 禁止的存储位置, 处理器将产生存储器管理故障, 导致故障异常, 并可能导致操作系统环境中的进程终止。在操作系统环境下, 内核可以动态更新 MPU 区域设置。在通常情况下, 嵌入式操作系统使用 MPU 保护内存。

(2) 嵌套向量中断控制器 (NVIC)

NVIC 的作用是实现低延迟中断处理。NVIC 对所有的异常和中断进行优先级划分和处理, 包括一个不可屏蔽中断 (Non Maskable Interrupt, NMI), 可以提供 256 个中断优先级。NVIC 与处理器内核紧密集成, 能够快速响应中断, 使得中断延迟很低。中断发生时处理器状态被自动存储到堆栈, 中断服务程序结束时又自动被恢复。向量的读取与状态保存并行, 使处理器高效率进入中断。称为尾链 (tail-chain) 的优化功能使得相邻的中断响应不需要重复的状态保存和恢复, 减少了相邻中断之间的切换时间。通过软件可设置 7 个异常 (系统处理) 和 96 个中断的 8 级优先级别。NVIC 操作具有如下特点。

- 可配置 1~240 的外部中断。
- 可配置 3~8 级的优先级。
- 动态优先级中断。
- 优先级分组, 可以选择优先中断级和非优先中断级。
- 确定而快速的中断处理, 只需要 12 个周期或 6 个周期 (相邻中断切换时)。
- 外部非屏蔽中断信号可用于安全的紧急应用。
- 通过硬件实现寄存器操作, 可处理特殊的中断。

(3) 系统控制模块 (SCB)

SCB 是编程模型与处理器之间的接口, 用于系统控制, 包括系统异常的配置、控制和报告等。

(4) 系统定时器 (SysTick)

SysTick 是一个 24 位减数 (递减到零再重装) 计数器, 可用作 RTOS 的节拍定时器或者一般的计数器。

(5) 单精度浮点运算单元 (FPU)

FPU 专门用来进行浮点运算, Cortex-M4 中的 Cortex-M4F 才有该模块, 具体特点如下。

- 具有 32 位单精度浮点运算指令。

- 集成了乘法和累加指令集，用于提高运算精度。
- 对于数据转换、加法、减法、乘法运算和可选的累加、除法和开方运算都有硬件支持。
- 通过硬件支持非正规的以及所有 IEEE 的舍入模式。
- 具有 32 个专用的 32 位单精度寄存器，同时也可当作 16 个 64 位双精度的寄存器使用。
- 采用解耦三级流水线结构。

2. 总线接口

总线包括 AHB-Lite 和 APB，AHB-Lite 总线的访问性能比 APB 高。3 条 AHB-Lite 总线分别为 ICode 总线、DCode 总线和系统总线（System Bus），ICode 总线和 DCode 总线分别用于从代码空间取指令和数据，系统总线则用于访问 SRAM 和其他外设。

私有外设总线（Private Peripheral Bus，PPB）基于高级外设总线（APB），包括内部和外部两条总线。

1) 内部 PPB 供以下设备使用。

- 仪表跟踪宏单元（ITM）。
- 数据观察点和跟踪（DWT）单元。
- Flash 修补和断点（FPB）。
- 系统控制空间（System Control Space，SCS），包括存储器保护单元（MPU）和嵌套向量中断控制器（NVIC）。

2) 外部 PPB 供以下设备使用。

- 跟踪端口接口单元（TPIU）。
- 嵌入式跟踪宏单元（ETM）。
- ROM 表。
- 实施特定区域 PPB 存储器映射。

除此之外，Cortex-M4 总线接口还有以下特点。

- 内存访问对齐。
- 支持位带（bit-band），包括位带原子的读写操作。
- 配有写缓冲区。
- 在多处理器系统中能单独访问。

3. 调试跟踪接口

调试跟踪接口提供低成本且功能强大的调试、跟踪和分析功能，具有如下特点。

1) 可以访问所有存储器、寄存器以及存储器映射的外设，当内核停止时可以访问内核寄存器。

2) 采用串行调试端口（Serial Wire Debug Port，SW-DP）或者串行 JTAG 调试端口（Serial Wire JTAG Debug Port，SWJ-DP）。

3) 可以访问以下设备。

- Flash 修补和断点（Flash Patch and Breakpoint，FPB），实现 Flash 断点设置和代码修补。
- 仪表跟踪宏单元（Instrumentation Trace Macrocell，ITM），实现 printf() 方式的调试。
- 数据观察点和跟踪（Data Watchpoint and Trace，DWT）单元，实现观察点和数据的跟踪，以及系统分析。

- 跟踪端口接口单元 (Trace Point Interface Unit, TPIU), 可以连接到跟踪端口分析器 (Trace Port Analyzer, TPA), 包括单线输出 (Single Wire Output, SWO) 模式。
- 嵌入式跟踪宏单元 (Embedded Trace Macrocell, ETM), 实现指令跟踪。

1.2.3 Cortex-M4 微处理器的编程模式

Cortex-M4 微处理器的编程模式是指处理器的运行模式, 包括以下两种。

- 1) 主模式 (Thread Mode)。用于执行应用软件的基本模式, 复位后处理器进入该模式。
- 2) 中断模式 (Handler Mode)。用于处理异常的模式, 所有异常在该模式下执行, 执行完成后返回主模式。

在处理器模式下, 软件执行有两种模式, 即特权或者非特权模式。特权模式可以执行所有指令并访问所有资源, 而非特权模式将不能或有限制地执行某些指令 (如 MSR、MRS、CPS) 及访问某些资源 (如系统时钟、NVIC、SCB、某些存储器及外设等)。主模式可以执行特权或者非特权模式, 中断模式则总是执行特权模式。

1.2.4 Cortex-M4 微处理器的堆栈

堆栈是一种寄存器的使用模型, 由一块连续的内存和一个堆栈指针组成, 用于实现“后进先出”的缓冲区。其典型应用是在发生中断时, 执行中断处理程序前后保护和恢复现场数据。对于具体的堆栈形式, 既可以“向上生长”, 又可以“向下生长”。

1. Cortex-M4 的堆栈

Cortex-M4 使用的是“向下生长”的堆栈模型, 即堆栈指针 SP 指向最后一个被压入堆栈的 32 位数值, 在下次压栈时, SP 先自动减 4, 再存入新的数值。初始化堆栈时, 堆栈指针 SP 指向的第一个地址, 叫作栈顶地址。

在 Cortex-M4 中, 堆栈指针的最低两位永远是 0, 这意味着堆栈总是 4 字节对齐的。

栈顶地址即堆栈大小的选择需要谨慎考虑。如果栈顶地址选择过低, 即堆栈较小, 则在中断嵌套大量消耗堆栈的情况下容易造成堆栈溢出, 进而可能改写其他内存数据, 出现不可预料的后果; 如果栈顶地址偏高, 即堆栈过大, 则容易造成内存浪费, 影响其他场合对内存的使用。

2. 双堆栈机制

在 Cortex-M4 中, 堆栈分为两个: 主堆栈 (Main Stack, MS) 和进程堆栈 (Process Stack, PS)。两个堆栈分别存储在内存中, 是两个不同的栈顶地址, 但在同一时刻只能使用其中一个, 不能同时使用。两个堆栈的选择是通过控制寄存器 (CONTROL) 的第 1 位, 即 CONTROL[1] 来决定的, 使用说明如下。

- 1) 当 CONTROL[1]=0 时, 只使用 MS, 这也是复位后的默认使用方式, 一般用于普通用户的程序堆栈。
- 2) 当 CONTROL[1]=1 时, 使用 PS, 一般供 OS (操作系统) 使用, 其作用是在使用 OS 的环境下防止用户的程序堆栈破坏 OS 使用的堆栈。

1.2.5 Cortex-M4 微处理器的内核寄存器

Cortex-M4 微处理器的内核寄存器如图 1-3 所示, 包括以下寄存器。

- 1) 13 个 32 位的通用寄存器 R0~R12。寄存器 R0~R7 可被所有指定了通用寄存器的指令访问；R8~R12 则只能被指定了通用寄存器的 32 位指令访问，不能被 16 位指令访问。
- 2) 堆栈指针寄存器（SP）即 R13。在线程模式下采取双堆栈机制。
- 3) 链接寄存器（LR）即 R14。用于从程序计数器 PC 接收跳转指令（BL、BLX）的程序返回地址，也可用于存储子程序、函数调用、中断等的返回信息，其他时间则可作为一般寄存器。
- 4) 程序计数器（PC）即 R15。用于保存当前程序地址。
- 5) 特殊用途的程序状态寄存器（xPSR）。用于保存当前程序的执行状态信息。

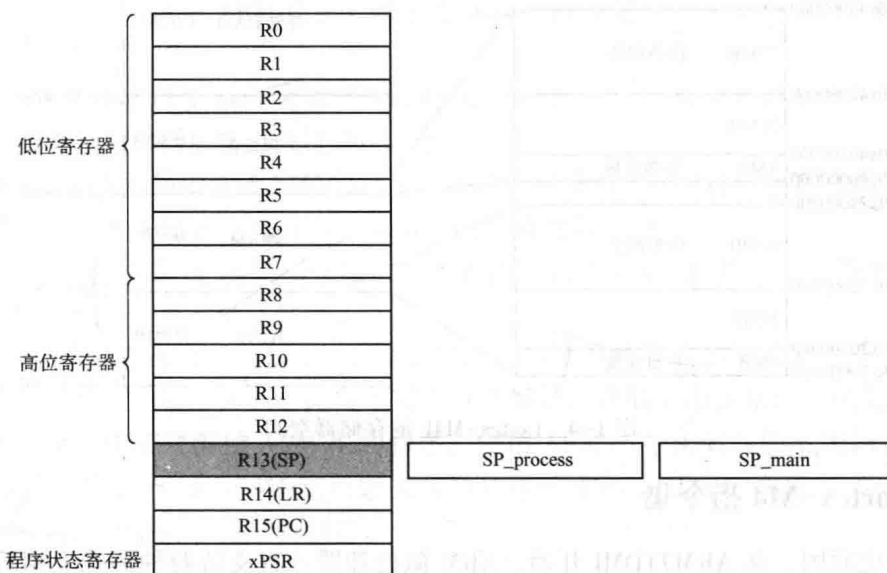


图 1-3 Cortex-M4 微处理器的内核寄存器

1.2.6 Cortex-M4 微处理器的存储器映射

Cortex-M4 微处理器的存储器空间大小为 4 GB，共分为 8 个区域，如图 1-4 所示。各区简介如下。

- 1) 代码区（Code）：大小为 0.5 GB，存储程序代码，指令通过 ICode 总线访问，数据通过 DCode 总线访问。
- 2) 内部 SRAM 区：大小为 0.5 GB，存储数据，指令和数据均通过系统总线访问。
- 3) 内核外设区（Peripheral）：大小为 0.5 GB，指令和数据均通过系统总线访问。
- 4) 外部 RAM 区（External RAM）：大小为 1 GB，指令和数据均通过系统总线访问。
- 5) 外部设备区（External device）：大小为 1 GB，指令和数据均通过系统总线访问。
- 6) 内部私有外设总线区（Private Peripheral Bus-Internal）：大小为 256 KB，通过内部 PPB 访问，该区域为不可执行区域（Execute Never，XN）。
- 7) 外部私有外设总线区（Private Peripheral Bus-External）：大小为 768 KB，通过外部 PPB 访问，该区域为不可执行区域。
- 8) 系统区（System）：大小为 511 MB，为器件制造商的系统外设区，该区域为不可执行区域。