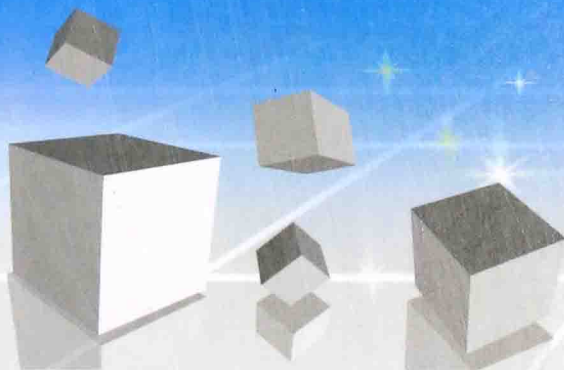
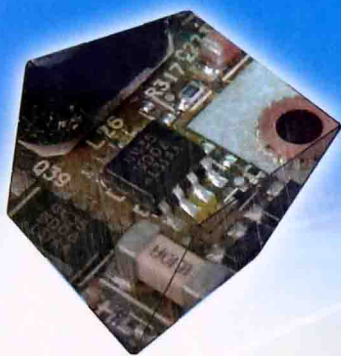




嵌入式技术与应用丛书  
ARM AAE认证考试教材



# ARM Cortex-M4 体系结构与外设接口实战开发

北京通联物联网教育咨询有限公司 奚海蛟 童 强 林庆峰 编著

- ◎ 详细介绍ARM Cortex-M4处理器的体系结构和外设接口
- ◎ 内容全面，讲解通俗易懂，适合读者学习使用
- ◎ 涵盖ARM AAE认证考试Cortex-M部分的内容，具有较强的针对性



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

014057433

TP332  
330

嵌入式技术与应用丛书  
ARM AAE 认证考试教材

# ARM Cortex-M4 体系结构 与外设接口实战开发

北京通联物联网教育咨询有限公司 奚海蛟 童 强 林庆峰 编著



TP332  
330

电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING



## 内 容 简 介

本书以 Cortex-M4 系列的 Kinetis 处理器为基础,从 ARM 的基础理论知识入手,以不同的外设为基准点讲解 Cortex-M4 系列的强大功能,主要内容涵盖 Cortex-M4 内核简介、Cortex-M4 内核结构、Cortex-M4 汇编指令集、DSP 扩展指令和 FPU、Cortex-M4 汇编语言与 C 语言混合编程、系统控制和存储管理、Cortex-M4 的异常处理、嵌套中断向量控制器、Cortex-M4 调试与跟踪系统、开发环境介绍、以  $\mu\text{C}/\text{OS-II}$  移植例程理解 Cortex-M4 内核编程、基于 Cortex-M4 内核的 Kinetis 架构介绍、Kinetis 系统控制模块和时钟模块、Freescale Kinetis 开发实例等内容。

本书适合参加 ARM AAE 认证考试的读者使用,也可供采用 ARM 平台进行开发的嵌入式从业人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

### 图书在版编目(CIP)数据

ARM Cortex-M4 体系结构与外设接口实战开发 / 奚海蛟, 童强, 林庆峰编著.

北京: 电子工业出版社, 2014.8

(嵌入式技术与应用丛书)

ISBN 978-7-121-23907-6

I. ①A… II. ①奚… ②童… ③林… III. ①微处理器—系统设计 IV. ①TP332

中国版本图书馆 CIP 数据核字(2014)第 170169 号

责任编辑: 田宏峰 特约编辑: 牛雪峰

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 21 字数: 530 千字

版 次: 2014 年 8 月第 1 版

印 次: 2014 年 8 月第 1 次印刷

印 数: 3000 册 定价: 59.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

ARM 是 32 位嵌入式微处理器行业的领先提供商，到目前为止，已推出各种各样基于通用体系结构的处理器，这些处理器具有高性能和行业领先的特点，而且系统成本也有所降低。ARM 与业界最广泛的体系（拥有超过 750 个可提供硅、工具和软件的合作伙伴）相结合，已推出的一系列 20 多种处理器，可以解决几乎每个应用方面的难题。迄今为止，ARM 已生产超过 200 亿个处理器，每天的销量超过 1 000 万，是真正意义上的 The Architecture for the Digital World（数字世界的体系结构）。

## 本书读者对象

本书以嵌入式 Cortex-M4 体系架构知识为主线，以飞思卡尔的 Kinetis 系列微控制器中的 K10 为蓝本，辅以大量的实战案例，系统地讲述了 Cortex-M4 系列框架结构及其开发流程。主要面向高年级本科生或研究生教学、有一定经验的嵌入式从业者，也可以作为培训机构的高级教材。

由于书中有大量 Kinetis K10 实战案例，而教育部支持的高校科技竞赛之一“全国大学生飞思卡尔杯智能汽车竞赛”从 2012 年起也可以使用 Kinetis 微控制器，一些参赛高校和个人也可以以此书作为参考资料。

2013 年 ARM 认证工程师（AAE）认证考试进入中国，ARM 公司授权北京通联物联网教育咨询有限公司为中国 AAE 认证考试培训中心，开展考前培训与考试组织工作，为此编写了一系列认证考试教材，本教材为 ARM Cortex-M 体系结构技术研发部分，既可以作为工程师参考工具书又可作为 AAE 考试基础教材。

AAE 考试报名与培训请联系：xhj1021@sina.com 或 QQ：2698408486。

## 本书主要内容

本书基于 Cortex-M4 架构知识框架编写，以飞思卡尔的 ARM Cortex-M4 内核的 Kinetis 系列微控制器为蓝本，阐述嵌入式为控制系统的软件和初级硬件设计。全书共 14 章：第 1 章为概述，简单阐述 ARM 公司的四大类微控制器和微处理器，包括特点和应用方向；第 2~9 章系统的阐述了 Cortex-M4 内核的知识体系，包括内核结构、汇编指令、DSP 和 FPU 指令、C 和汇编混合编程、系统控制和存储管理、异常和中断、调试与跟踪系统；第 10 章为开发环境介绍，介绍 Cortex-M4 内核家族的微控制器的几种开发环境介绍，包括 CodeWarrior、Keil、IAR 等主流开发环境，并一步一步地带领读者创建了第一个工程，本书使用 CodeWarrior 作为案例开发环境；第 11~14 章是实战部分，选用飞思卡尔公司基于 Cortex-M4 内核的 Kinetis 系列微控制器 K10 为实战硬件，包括  $\mu$ C/OS-II 在 K10 上的移植、Kinetis 架构的详细介绍、Kinetis 系列的系统控制和时钟模块理论和示例，尤为突出的是，在第 14 章，详细地编写了 GPIO、RTC、UART、PIT 定时器、Watchdog、ADC、GPIO 的矩阵键盘应用、PWM 八大模块的理论和实战示例（其中矩阵键盘和 PWM 模块用到了外接的硬件，K10 核心板上没有这些硬件模块），让读者真正地领会 Kinetis 系列微控制器的编程流程和每一个细节。

## 本书随书资料

读者可以在 <http://dl.vmall.com/c0x4knzeev> 下载本书的全部案例代码，也可以登录华信教育资源网 ([www.hxedu.com.cn](http://www.hxedu.com.cn)) 免费注册后下载。书中第 14 章涉及的 K10 核心开发板、USBDM 调试器、矩阵键盘、舵机等硬件可以使读者更清晰把握本书的基础，如有需要读者可以联系 [xhj1021@sina.com](mailto:xhj1021@sina.com) 或 QQ (2698408486) 购买。

参与本书编写工作的还有苗凤娟、乔林、陈晓冬、谌利、李政春、刘张辉、李晓庆、滕忠楠、付盈、王秀文、吴飞、王丽娜。

鉴于作者水平有限，书中难免存在不足和错误之处，恳请读者提出宝贵意见和建议，以便再版时改进。

编 者

2014 年 7 月

# 目 录

|       |                         |    |
|-------|-------------------------|----|
| 第 1 章 | Cortex-M4 内核简介          | 1  |
| 1.1   | 主流的 ARM 处理器介绍           | 1  |
| 1.1.1 | ARM 体系结构的命名规则           | 2  |
| 1.1.2 | ARM Cortex 应用程序处理器      | 2  |
| 1.1.3 | ARM Cortex 嵌入式处理器       | 4  |
| 1.1.4 | 经典 ARM 处理器              | 8  |
| 1.1.5 | ARM 专家处理器               | 9  |
| 1.1.6 | 其他 ARM 处理器              | 10 |
| 1.1.7 | ARM 处理器的特点              | 11 |
| 1.2   | 应用领域                    | 11 |
| 1.3   | ARM Cortex-M4 简介        | 12 |
| 1.3.1 | Cortex-M4 信号处理技术        | 13 |
| 1.3.2 | ARM Cortex-M 微控制器技术     | 13 |
| 1.4   | 以 ARM Cortex-M4 为核心的处理器 | 14 |
|       | 本章小结                    | 16 |
| 第 2 章 | Cortex-M4 内核结构          | 17 |
| 2.1   | Cortex-M4 处理器与核心外围设备    | 17 |
| 2.2   | 流水线                     | 21 |
| 2.3   | Cortex-M4 的总线接口         | 23 |
| 2.4   | Cortex-M4 相关寄存器组        | 28 |
| 2.4.1 | 寄存器                     | 28 |
| 2.4.2 | 异常与中断                   | 34 |
| 2.5   | 操作模式                    | 34 |
|       | 本章小结                    | 35 |
| 第 3 章 | Cortex-M4 汇编指令集         | 37 |
| 3.1   | Cortex-M4 指令集           | 37 |
| 3.1.1 | ARM32 和 Thumb 指令集       | 37 |
| 3.1.2 | Thumb-2 指令集             | 38 |
| 3.2   | CMSIS 标准                | 38 |
| 3.3   | ARM 汇编语言基础              | 39 |
| 3.3.1 | 指令的格式                   | 40 |
| 3.3.2 | 操作数                     | 40 |
| 3.3.3 | PC 与 SP 寄存器的使用限制        | 40 |
| 3.3.4 | 灵活的第二个操作数               | 40 |
| 3.3.5 | 移位运算                    | 40 |

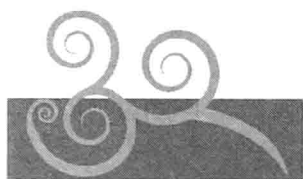
|        |                         |    |
|--------|-------------------------|----|
| 3.3.6  | 地址对齐                    | 43 |
| 3.3.7  | 标号——PC 相对地址表达式          | 44 |
| 3.3.8  | 条件码                     | 44 |
| 3.3.9  | 指令宽度                    | 46 |
| 3.4    | Cortex-M4 处理器寻址方式       | 47 |
| 3.4.1  | 立即寻址                    | 47 |
| 3.4.2  | 寄存器寻址                   | 48 |
| 3.4.3  | 寄存器间接寻址                 | 48 |
| 3.4.4  | 寄存器偏移寻址                 | 48 |
| 3.4.5  | 基址变址寻址                  | 48 |
| 3.4.6  | 多寄存器寻址                  | 49 |
| 3.4.7  | 堆栈寻址                    | 49 |
| 3.5    | Cortex-M4 指令集           | 50 |
| 3.5.1  | 分支指令                    | 50 |
| 3.5.2  | 常见的数据处理指令               | 55 |
| 3.5.3  | 乘除指令                    | 59 |
| 3.5.4  | 饱和指令                    | 61 |
| 3.5.5  | 扩展指令                    | 62 |
| 3.5.6  | 位段操作指令                  | 63 |
| 3.5.7  | 状态寄存器访问指令               | 65 |
| 3.5.8  | 加载存储指令                  | 67 |
| 3.5.9  | 多寄存器加载和存储指令             | 69 |
| 3.5.10 | 杂项处理指令                  | 70 |
| 3.5.11 | 异常生成指令                  | 72 |
| 3.5.12 | 协处理指令——Cortex-M4 不支持的指令 | 73 |
| 3.6    | 汇编实例                    | 74 |
|        | 本章小结                    | 75 |
| 第 4 章  | DSP 扩展指令和 FPU           | 77 |
| 4.1    | 简介                      | 77 |
| 4.2    | DSP 扩展指令                | 77 |
| 4.2.1  | 乘加指令                    | 78 |
| 4.2.2  | 饱和指令                    | 82 |
| 4.2.3  | 位段指令                    | 83 |
| 4.2.4  | SIMD 并行加减指令             | 85 |
| 4.2.5  | 杂项数据处理指令                | 86 |
| 4.3    | FPU                     | 87 |
| 4.3.1  | FPU 的寄存器                | 87 |
| 4.3.2  | FPU 的操作模式               | 88 |
| 4.3.3  | 浮点数加载存储指令               | 88 |
| 4.3.4  | 浮点数寄存器传送指令              | 89 |

|                                       |            |
|---------------------------------------|------------|
| 4.3.5 浮点数数据处理指令                       | 89         |
| 本章小结                                  | 91         |
| <b>第 5 章 Cortex-M4 汇编与 C 混合编程</b>     | <b>93</b>  |
| 5.1 ATPCS 概述                          | 93         |
| 5.1.1 寄存器的使用规则                        | 94         |
| 5.1.2 数据栈使用规则                         | 94         |
| 5.1.3 参数传递规则                          | 95         |
| 5.2 内嵌汇编                              | 96         |
| 5.2.1 内嵌汇编的语法格式                       | 96         |
| 5.2.2 内嵌汇编指令的特点                       | 96         |
| 5.2.3 内嵌汇编指令注意事项                      | 98         |
| 5.2.4 内嵌汇编指令举例                        | 100        |
| 5.3 ARM 中的汇编和 C 语言相互调用                | 101        |
| 5.3.1 从汇编程序中访问 C 程序变量                 | 102        |
| 5.3.2 C 程序调用汇编程序                      | 102        |
| 5.3.3 汇编程序调用 C 程序                     | 104        |
| 本章小结                                  | 104        |
| <b>第 6 章 系统控制和存储管理</b>                | <b>105</b> |
| 6.1 系统控制                              | 105        |
| 6.1.1 系统控制和 ID 寄存器                    | 106        |
| 6.1.2 系统定时器                           | 107        |
| 6.1.3 可嵌套中断向量控制器 (NVIC)               | 108        |
| 6.1.4 内存保护系统架构 (PMSAv7)               | 109        |
| 6.2 内存映射                              | 109        |
| 6.2.1 对齐访问                            | 112        |
| 6.2.2 字节序 (Endian)                    | 113        |
| 6.3 位带操作                              | 114        |
| 6.3.1 bit-band 和 bit-band alias 区域的访问 | 116        |
| 6.3.2 bit-banding 的功能                 | 116        |
| 6.4 内存保护单元 (MPU)                      | 117        |
| 6.4.1 内存保护单元 (MPU) 简介                 | 117        |
| 6.4.2 MPU 寄存器组                        | 119        |
| 6.4.3 MPU 应用示例                        | 122        |
| 本章小结                                  | 123        |
| <b>第 7 章 Cortex-M4 的异常处理</b>          | <b>125</b> |
| 7.1 异常类型                              | 125        |
| 7.2 异常的优先级                            | 127        |
| 7.3 异常向量表                             | 130        |
| 7.4 异常处理                              | 131        |

|        |                                         |     |
|--------|-----------------------------------------|-----|
| 7.4.1  | 异常的进入 .....                             | 131 |
| 7.4.2  | 异常中断的响应时序 .....                         | 132 |
| 7.4.3  | 异常的返回 .....                             | 133 |
| 7.4.4  | 中断 late-arrive 和 tail-chaining 机制 ..... | 133 |
| 7.5    | 故障类异常 .....                             | 135 |
| 7.5.1  | 总线故障 .....                              | 135 |
| 7.5.2  | 存储器管理故障 .....                           | 136 |
| 7.5.3  | 应用故障 .....                              | 138 |
| 7.5.4  | 硬故障 .....                               | 139 |
|        | 本章小结 .....                              | 139 |
| 第 8 章  | 嵌套中断向量控制器 .....                         | 141 |
| 8.1    | NVIC 介绍 .....                           | 141 |
| 8.2    | NVIC 工作原理 .....                         | 142 |
| 8.3    | NVIC 寄存器 .....                          | 143 |
| 8.4    | 中断编程实例 .....                            | 145 |
|        | 本章小结 .....                              | 151 |
| 第 9 章  | Cortex-M4 调试与跟踪系统 .....                 | 153 |
| 9.1    | 嵌入式调试技术概述 .....                         | 153 |
| 9.2    | CoreSight 技术介绍 .....                    | 154 |
| 9.3    | Cortex-M4 调试架构 .....                    | 155 |
| 9.4    | SW-DP 和 SWJ-DP .....                    | 156 |
| 9.4.1  | SWJ-DP .....                            | 156 |
| 9.4.2  | SWJ-DP 接口 .....                         | 157 |
| 9.4.3  | SWJ-DP 中 JTAG 和 SWD 的选择机制 .....         | 157 |
| 9.5    | Cortex-M4 调试模式 .....                    | 158 |
| 9.6    | Cortex-M4 的跟踪系统 .....                   | 160 |
| 9.6.1  | 数据观察点与跟踪 (DWT) .....                    | 161 |
| 9.6.2  | 仪器化跟踪宏单元 (ITM) .....                    | 162 |
| 9.6.3  | 嵌入式跟踪单元 ETM .....                       | 163 |
| 9.6.4  | 跟踪端口接口单元 (TPIU) .....                   | 163 |
| 9.6.5  | 闪存地址重载及断点单元 (FPB) .....                 | 164 |
|        | 本章小结 .....                              | 165 |
| 第 10 章 | 开发环境介绍 .....                            | 167 |
| 10.1   | USBDM 调试器 .....                         | 167 |
| 10.2   | CodeWarrior for MCU v10.2 介绍 .....      | 168 |
| 10.2.1 | CodeWarrior 的功能和特点 .....                | 168 |
| 10.2.2 | CodeWarrior 安装与配置 .....                 | 168 |
| 10.2.3 | 建立第一个工程 .....                           | 169 |
| 10.3   | Keil MDK 介绍 .....                       | 171 |

|        |                                        |     |
|--------|----------------------------------------|-----|
| 10.4   | Keil ULINK 仿真器 .....                   | 173 |
| 10.5   | MDK-ARM 的软件开发周期 .....                  | 174 |
| 10.6   | μVisionIDE 的使用 .....                   | 174 |
| 10.6.1 | μVision 调试器 .....                      | 176 |
| 10.6.2 | 建立第一个工程 .....                          | 177 |
| 10.6.3 | 调试 .....                               | 180 |
| 10.6.4 | 烧写 Flash .....                         | 185 |
| 10.7   | IAR for ARM 基本介绍 .....                 | 186 |
| 10.7.1 | IAR EWARM 开发环境 IDE 介绍 .....            | 187 |
| 10.7.2 | 建立第一个工程 .....                          | 188 |
| 10.7.3 | 调试器 C-SPY .....                        | 191 |
|        | 本章小结 .....                             | 193 |
| 第 11 章 | 以 μC/OS-II 移植例程理解 Cortex-M4 内核编程 ..... | 195 |
| 11.1   | μC/OS-II 操作系统 .....                    | 195 |
| 11.1.1 | μC/OS-II 操作系统简介 .....                  | 195 |
| 11.1.2 | μC/OS-II 的工作原理 .....                   | 196 |
| 11.2   | μC/OS-II 在 Kinetis K10 开发板上的移植 .....   | 196 |
| 11.2.1 | μC/OS-II 目录结构简介 .....                  | 196 |
| 11.2.2 | 移植概述 .....                             | 197 |
| 11.2.3 | 移植详解 .....                             | 198 |
| 11.3   | 测试 μC/OS-II .....                      | 209 |
| 11.4   | 板级支持包 .....                            | 214 |
|        | 本章小结 .....                             | 214 |
| 第 12 章 | 基于 Cortex-M4 内核的 Kinetis 架构介绍 .....    | 215 |
| 12.1   | Cortex-M4 总线接口 .....                   | 215 |
| 12.2   | Cortex-M4 内核同 Kinetis 系统架构的连接 .....    | 216 |
| 12.3   | 交叉开关 .....                             | 218 |
| 12.3.1 | 交叉开关简介 .....                           | 218 |
| 12.3.2 | 交叉开关的寄存器定义 .....                       | 218 |
| 12.3.3 | 交叉开关功能描述 .....                         | 219 |
| 12.3.4 | Kinetis K10 处理器中的交叉开关 .....            | 219 |
| 12.3.5 | Kinetis K10 处理器中的 MPU .....            | 220 |
|        | 本章小结 .....                             | 221 |
| 第 13 章 | Kinetis 系统控制模块和时钟模块 .....              | 223 |
| 13.1   | Kinetis 的系统控制 .....                    | 223 |
| 13.1.1 | Kinetis 架构 .....                       | 223 |
| 13.1.2 | 核心总线 .....                             | 224 |
| 13.1.3 | 交叉开关 .....                             | 224 |
| 13.1.4 | 内存耦合 .....                             | 226 |
| 13.1.5 | 系统访问时间 .....                           | 228 |

|        |                       |     |
|--------|-----------------------|-----|
| 13.2   | 时钟管理                  | 229 |
| 13.2.1 | 多时钟发生器                | 229 |
| 13.2.2 | 系统振荡器                 | 235 |
| 13.2.3 | 实时时钟振荡器               | 238 |
| 13.3   | 电源管理                  | 238 |
| 13.3.1 | 电源模式                  | 239 |
| 13.3.2 | 进入及退出电源模式             | 240 |
| 13.3.3 | 电源模式转换                | 240 |
| 13.3.4 | 电源模式关闭序列              | 240 |
| 13.3.5 | 低功耗模式操作模块             | 241 |
| 13.3.6 | 时钟门控技术                | 244 |
| 13.3.7 | 时钟初始化实例               | 244 |
|        | 本章小结                  | 248 |
| 第 14 章 | Freescal Kinetis 开发实例 | 249 |
| 14.1   | 系统初始化框架               | 249 |
| 14.2   | GPIO                  | 257 |
| 14.3   | 实时时钟 RTC              | 266 |
| 14.4   | 通用异步收发器               | 272 |
| 14.4.1 | UART 功能概述             | 272 |
| 14.4.2 | UART 相关寄存器            | 273 |
| 14.4.3 | UART 测试实例             | 278 |
| 14.5   | 定时器                   | 287 |
| 14.5.1 | 定时器简介                 | 287 |
| 14.5.2 | 定时器测试实例               | 288 |
| 14.6   | 看门狗                   | 291 |
| 14.6.1 | 看门狗简介                 | 291 |
| 14.6.2 | 测试实例                  | 293 |
| 14.7   | 模/数转换模块               | 295 |
| 14.7.1 | ADC 功能概述              | 295 |
| 14.7.2 | ADC 特殊功能寄存器           | 296 |
| 14.7.3 | ADC 测试实例              | 298 |
| 14.8   | 矩阵键盘                  | 300 |
| 14.8.1 | 键盘概述                  | 300 |
| 14.8.2 | 矩阵键盘测试实例              | 302 |
| 14.9   | PWM 模块操作 (servo9g 舵机) | 309 |
| 14.9.1 | FTM 功能概述              | 309 |
| 14.9.2 | FTM 特殊功能寄存器           | 310 |
| 14.9.3 | FTM 控制舵机测试实例          | 318 |
|        | 本章小结                  | 324 |
|        | 参考文献                  | 325 |



# 第 1 章

## Cortex-M4 内核简介

本章主要介绍了 ARM 处理器的特点以及它们的应用领域，并详细介绍了以 ARM Cortex-M4 为核心的处理器。

本章要点：

- 主流的 ARM 处理器介绍。
- ARM 处理器及技术的应用领域。
- ARM Cortex-M4 处理器简介。



### 1.1 主流的 ARM 处理器介绍

ARM 是 32 位嵌入式微处理器行业的领先提供商，到目前为止，已推出各种各样基于通用体系结构的处理器，这些处理器具有高性能和行业领先的特点，而且系统成本也有所降低。ARM 与业界最广泛的体系（拥有超过 750 个可提供硅、工具和软件的合作伙伴）相结合，已推出的一系列 20 多种处理器可以解决几乎每个应用方面的难题。迄今为止，ARM 已生产超过 200 亿个处理器，每天的销量超过 1 000 万，是真正意义上的 The Architecture for the Digital World（数字世界的体系结构）。

目前，ARM 微处理器宏伟而丰富的体系主要包括以下四个部分：

- ARM Cortex 应用程序处理器。
- ARM Cortex 嵌入式处理器。
- 经典 ARM 处理器。
- ARM 专家处理器。

图 1.1 直观地展示了 ARM 不同系列处理器在性能、功能和兼容性之间的关系。

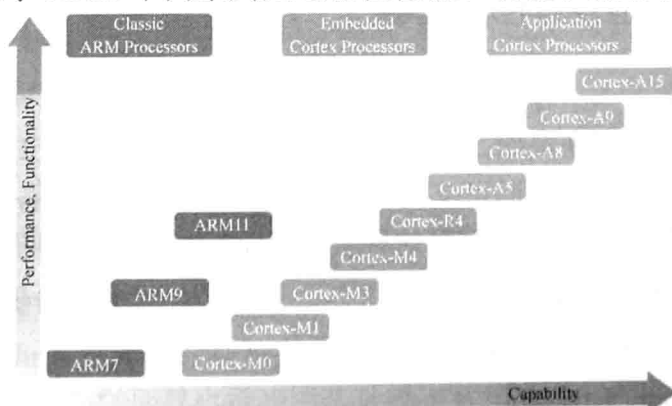


图 1.1 ARM 不同系列处理器之间关系

1.1.1 ARM 体系结构的命名规则

为便于理解 ARM 的各个型号的命名含义，也便于根据设计的功能进行芯片的选型，在详细介绍各个系列的 ARM 处理器之前，先补充一下 ARM 命名规则的知识。

ARM 体系结构是 CPU 产品所使用的一种体系结构，ARM 公司开发了一套拥有知识产权的 RISC 体系结构的指令集。每个 ARM 处理器都有一个特定的指令集架构，而一个特定的指令集架构又可以由多种处理器实现。特定的指令集架构随着嵌入式市场的发展而发展，由于所有产品均采用一个通用的软件体系，所以相同的软件可在所有产品中运行（在理论上如此）。

ARM 产品通常以 ARM 【x】【y】【z】【T】【D】【M】【I】【E】【J】【F】【-S】形式出现。表 1-1 显示了 ARM 体系结构的命名规则中这些后缀的具体含义。

表 1-1 ARM 体系结构的命名规则

| 后缀变量 | 含 义                    | 后缀变量 | 含 义                 |
|------|------------------------|------|---------------------|
| x    | 系列号，如 ARM7、ARM9 等      | y    | 存储管理/保护单元           |
| z    | Cache                  | T    | 支持 16 位压缩指令集 Thumb  |
| D    | 拥有 JTAG 调试器，支持片上 Debug | M    | 内嵌硬件乘法器（Multiplier） |
| I    | 嵌入式 ICE，支持片上断点和调试点     | E    | 增强指令（基于 TDMI）       |
| J    | Jazelle 加速             | F    | 向量浮点单元              |
| -S   | 可综合版本                  |      |                     |

此外，还有一些附加的要点：

- （1）ARM7TDMI 之后的所有 ARM 内核，即使“ARM”标志后没有包含“TDMI”字符，也都默认包含了 TDMI 的功能特性。
- （2）JTAG 是由 IEEE 1149.1 标准测试访问端口和边界扫描结构来描述的，它是 ARM 用来发送和接收处理器内核与测试仪器之间调试信息的一系列协议。
- （3）嵌入式 ICE 宏单元是建立在处理器内部用来设置断点和观察点的调试硬件。
- （4）可综合，意味着处理器内核是以源代码形式提供的，这种源代码形式可被编译成一种易于 EDA 工具使用的形式。

1.1.2 ARM Cortex 应用程序处理器



作为目前 ARM 处理器中最高端的系列，Cortex 应用程序处理器在高级工艺节点中已经可实现高达 2 GHz 的标准频率的卓越性能，从而可支持下一代的移动 Internet 设备。这些处理器具有单核和多核等不同的种类，最多可提供四个具有可选 NEON 多媒体处理模块和高级浮点执行单元的处理单元。

应用程序处理器按处理器执行复杂操作系统（如 Linux、Android/Chrome、Microsoft Windows（CE/嵌入式）和 Symbian）及支持复杂图形用户界面的能力来定义，此类处理器集成了内存管理单元（MMU）以管理这些复杂操作系统的内存需求，并允许下载和执行第三方软件。

ARM Cortex 应用程序处理器主要包含 Cortex-A 系列——开放式操作系统的高性能处理器，越来越接近 PC 了。下面来看看 Cortex-A 处理器卓越的性能表现。

## 1. Cortex-A 处理器概述

ARM Cortex-A 系列的应用型处理器可向托管丰富的操作系统平台的设备和用户应用提供全方位的解决方案，包括超低成本的手机、智能手机、移动计算平台、数字电视、机顶盒、企业网络、打印机和服务器解决方案。高性能的 Cortex-A15、可伸缩的 Cortex-A9、经过市场验证的 Cortex-A8 处理器和高效的 Cortex-A5 处理器均共享同一体系结构，因此具有完整的应用兼容性，支持传统的 ARM、Thumb 指令集和新增的高性能紧凑型 Thumb-2 指令集。

## 2. Cortex-A 处理器的优势

- (1) 移动 Internet 的理想选择。
- (2) 本身支持 Adobe Flash 10.1。
- (3) 高性能 NEON 引擎，广泛支持媒体编/解码器。
- (4) 低功耗设计，支持全天浏览和连接。
- (5) 高性能。Cortex-A 设备可为其目标应用领域提供各种可伸缩的能效性能点。一些说明示例如下：

- Cortex-A15 可为新一代移动基础结构应用和要求苛刻的无线基础结构应用提供性能最高的解决方案。
  - Cortex-A9 可提供 800 MHz~2 GHz 的标准频率，每个内核可提供 5000DMIPS 的性能。
  - Cortex-A8 单核解决方案，可提供经济有效的高性能，在 600 MHz~1 GHz 的频率下，提供的性能超过 2000DMIPS。
  - Cortex-A5 低成本实现，在 400~800 MHz 的频率下，提供的性能超过 1200DMIPS。
- (6) 多核技术。Cortex-A5、Cortex-A9 和 Cortex-A15 处理器都支持 ARM 的第二代多核技术。
    - 单核到四核实现，支持面向性能的应用领域。
    - 支持对称和非对称的操作系统实现。
    - 通过加速器一致性端口（ACP）再导出到系统的整个处理器中保持一致性。
  - (7) 高级扩展。除了具有与上一代经典 ARM 和 Thumb 体系结构的二进制兼容性外，Cortex-A 类处理器还通过以下技术扩展具有了更大优势。

- Thumb-2：提供最佳代码大小和性能。
- TrustZone 安全扩展：提供可信计算。
- Jazelle 技术：提高执行环境（如 Java、Net、MSIL、Python 和 Perl）速度。

## 3. Cortex-A 处理器的应用场合

Cortex-A 处理器适用于具有高计算要求、运行丰富的操作系统以及提供交互媒体和图形体验的应用领域。从最新技术的移动 Internet 必备设备（如手机和超便携的上网本或智能本）到汽车信息娱乐系统和下一代数字电视系统。

虽然 Cortex-A 处理器正朝着提供完全的 Internet 体验的方向发展，但其应用也很广泛，如表 1-2 所示。

表 1-2 ARM Cortex-A 处理器应用

| 产 品 类 型 | 应 用                       |
|---------|---------------------------|
| 计算      | 上网本、智能本、输入板、电子书阅读器        |
| 手机      | 智能手机、特色手机                 |
| 数字家电    | 机顶盒、数字电视、蓝光播放器、游戏控制台      |
| 汽车      | 信息娱乐、导航                   |
| 企业      | 激光打印机、路由器、无线基站、VOIP 电话和设备 |
| 无线基础结构  | Web2.0、无线基站、交换机、服务器       |

1.1.3 ARM Cortex 嵌入式处理器

区别于 Cortex 应用程序处理器对操作系统和移动互联网的支持偏向，Cortex 嵌入式处理器旨在为各种不同的市场提供服务。嵌入式处理器主要着重于在各种功耗敏感型应用中提供具有高确定性的实时行为，这些处理器通常执行实时操作系统（RTOS）和用户开发的应用程序代码，因此只需内存保护单元（MPU），不需应用程序处理器中提供的 MMU。

- ARM Cortex 嵌入式处理器主要包括 Cortex-M 系列和 Cortex-R 系列。
- （1）Cortex-M 系列：面向具有确定性的微控制器应用的成本敏感型解决方案，单片机风格的系统，这是本书重点要讲的。
  - （2）Cortex-R 系列：面向实时应用的卓越性能，既要快速又要实时。

从上面的简单介绍可以明显看出 M 和 R 系列的市场定位的区别，在图 1.1 中也能看到 R 系列在实时性等性能方面要优于 M 系列，而 M 系列在成本上更具优势。下面具体介绍两种系列的处理器。

1. Cortex-M 处理器

ARM Cortex-M 系列处理器主要是针对微控制器领域开发的，在该领域中，既需进行快速且具有高确定性的中断管理，又需将门数和可能功耗控制在最低。

Cortex-M 处理器是一系列可向上兼容的高能效、易于使用的处理器，这些处理器旨在帮助开发人员满足将来的嵌入式应用的需要。这些需要包括以更低的成本提供更多功能、不断增加连接、改善代码重用和提高能效。

简单地说，ARM Cortex-M 处理器提供优于 8 位和 16 位体系结构的代码密度，这在减少对内存的需求和最大限度地提高宝贵的芯片上闪存的使用率方面具有很大的优势。随着技术和应用要求的不断更新，对不断增加连接（如 USB、蓝牙和 IEEE 802.15）、具有复杂模拟传感器（如加速计和触摸屏）且成本日益降低的产品的需求，已导致需要将模拟设备与数字功能更紧密地集成，以对数据进行预处理和传输。而问题是，大多数 8 位设备在不显著增加频率（并因此不显著增加功率）的情况下不提供支持这些任务的性能，因此嵌入式开发人员需要寻找具有更高级处理器技术的替代设备。16 位设备以前曾被用来解决微控制器应用中的能效问题，但是 16 位设备相对不高的性能意味着它们通常需要较长的活动工作周期或较高的时钟频率才能完成

32 位设备所完成的相同任务。因此, Cortex 嵌入式处理器就为解决这些应用要求提供了合适的方案, 而 M 系列在整个 Cortex 处理器中的价格优势更为降低成本给出了一个很好的选择。

当然, 除了指令执行效率的优势之外, 基于 ARM Cortex 处理器的微控制器的软件开发可能比 8 位微控制器产品的开发容易得多。Cortex 处理器不但是完全可通过 C 语言进行编程的, 而且还附带各种高级调试功能以帮助定位软件中的问题。除 MCU 开发工具包中包括的所有附加资源外, 网上还有大量示例和教程, 其中许多来自基于 ARM 处理器的 MCU 供应商的网站。

Cortex-M 系列针对成本和功耗敏感的 MCU 与终端应用(如智能测量、人机接口设备、汽车和工业控制系统、大型家用电器、消费性产品和医疗器械)的混合信号设备进行过优化。总之, Cortex-M 系列是必须考虑不同的成本、能耗和性能的各类可兼容、易于使用的嵌入式设备(如微控制器)的理想解决方案。

(1) Cortex-M 的特点如下。

- 更低的功耗: 更低的频率或更短的活动时段运行, 基于架构的睡眠模式支持, 比 8/16 位设备的工作方式更智能, 睡眠时间更长。
- 更小的代码(更低的硅成本): 高密度指令集, 比 8/16 位设备每字节完成更多操作, 更小的 RAM、ROM 或闪存要求。
- 易于使用: 多个供应商之间的全球标准, 代码兼容性, 统一的工具和操作系统支持。
- 更有竞争力的产品: Powerful Cortex-M processor, 每 MHz 提供更高的性能, 能够以更低的功耗实现更丰富的功能

(2) Cortex-M 系列处理器如图 1-2 所示。

ARM Cortex-M0 处理器是现有的最小、能耗最低和能效最高的 ARM 处理器, 见图 1-2(a)。该处理器硅面积积极小、能耗极低并且所需的代码量极少, 这使得开发人员能够以 8 位的设备实现 32 位设备的性能, 从而省略 16 位设备的研发步骤。Cortex-M0 处理器超低的门数也使得它可以部署在模拟和混合信号设备中。

ARM Cortex-M1 处理器是第一个专为 FPGA 实现设计的 ARM 处理器, 见图 1-2(b)。Cortex-M1 处理器面向所有主要 FPGA 设备并包括对领先的 FPGA 综合工具的支持, 允许设计者为每个项目选择最佳实现。Cortex-M1 处理器使 OEM 能够通过跨 FPGA、ASIC 和 ASSP 的多个项目之间合理地利用软件和工具投资来节省大量成本, 此外还能够通过使用行业标准处理器实现更大的供应商独立性。

ARM Cortex-M3 处理器是行业领先的 32 位处理器, 适用于具有高确定性的实时应用, 已专门开发为允许合作伙伴为范围广泛的设备(包括微控制器、汽车车体系统、工业控制系统以及无线网络和传感器)开发高性能低成本的平台, 见图 1-2(c)。该处理器提供出色的计算性能和对事件的卓越系统响应, 同时可以应对低动态和静态功率限制的挑战。该处理器是高度可配置的, 可以支持范围广泛的实现(从那些需要内存保护和强大跟踪技术的实现到那些需要极小面积的对成本非常敏感的设备)。

本书主要以 ARM Cortex-M4 处理器进行讲解, 1.3 节会有详细的介绍。

(3) Cortex-M 处理器的兼容性。Cortex-M 系列是必须考虑不同的成本、能耗和性能的各类可兼容、易于使用的嵌入式设备(如微控制器)的理想解决方案。每个处理器都针对十分广泛的嵌入式应用范围提供最佳权衡取舍, 如表 1-3 所示。

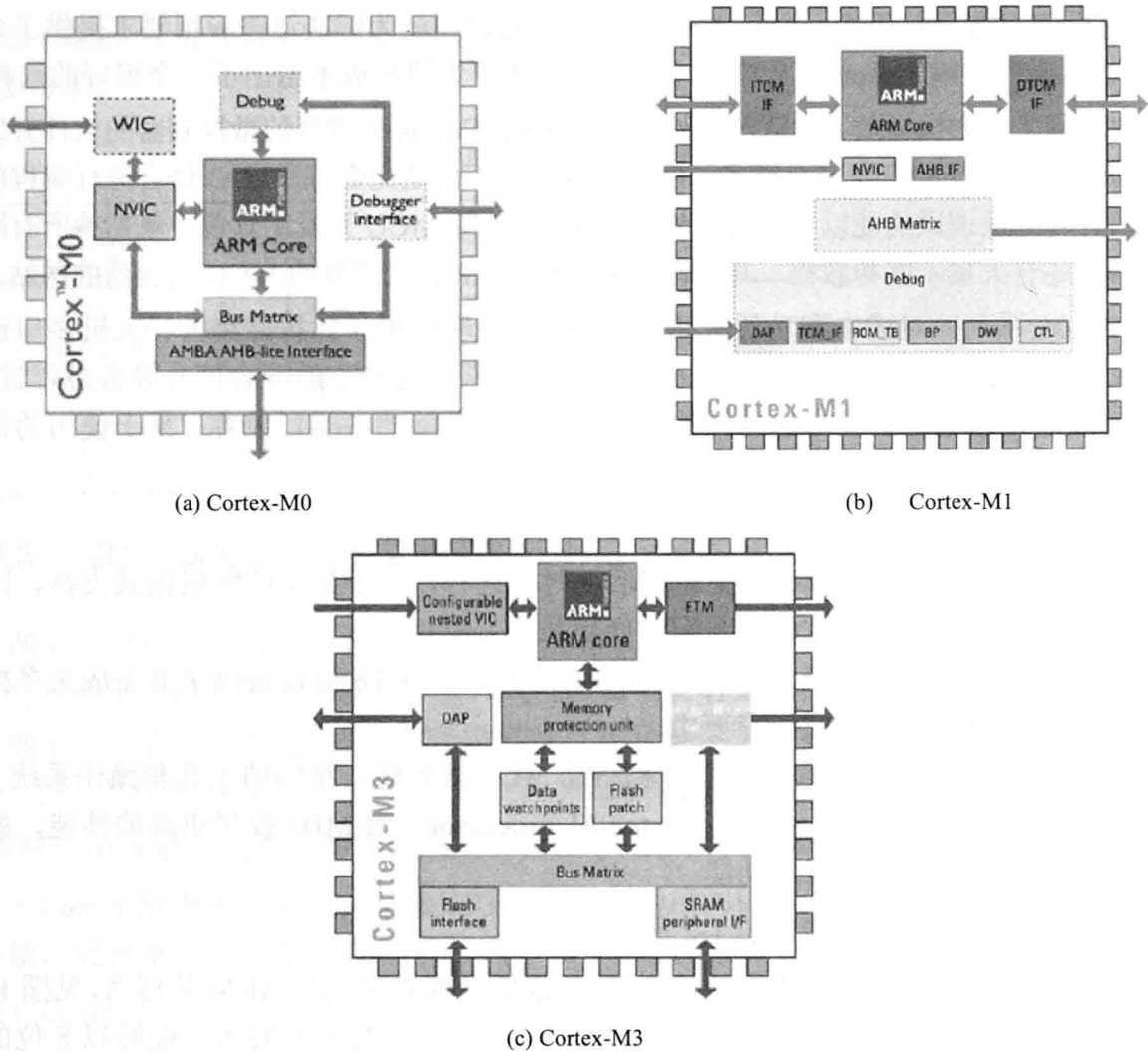


图 1.2 Cortex-M 系列处理器

表 1-3 Cortex-M 处理器

| ARM Cortex-M0 | ARM Cortex-M3 | ARM Cortex-M4 |
|---------------|---------------|---------------|
| 8/16 位应用      | 16/32 位应用     | 32 位/DSC 应用   |
| 低成本和简单性       | 性能效率          | 有效的数字信号控制     |

Cortex-M 系列处理器都是二进制向上兼容的，这使得软件重用，以及从一个 Cortex-M 处理器无缝发展到另一个成为可能，如图 1-3 所示。

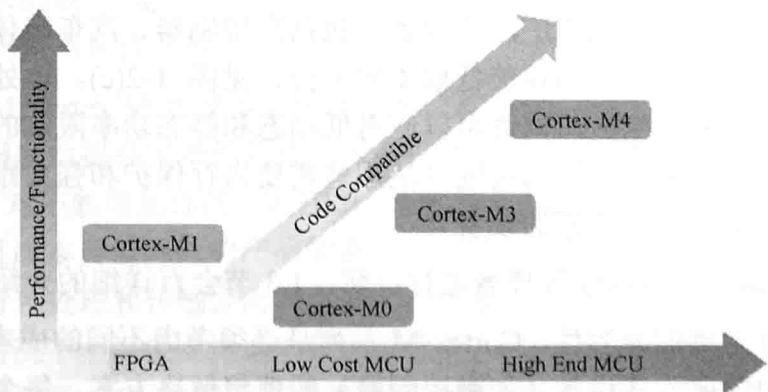


图 1.3 Cortex-M 处理器的兼容性