



ARM嵌入式设计与Windows CE 开发系列丛书

# ARM 原理与 C 程序设计

ARM Theory & C Program Design

张 勇 编著



西安电子科技大学出版社

<http://www.xduph.com>

ARM 嵌入式设计与 Windows CE 开发系列丛书

# ARM原理与C程序设计

张 勇 编著

西安电子科技大学出版社

2009

## 内 容 简 介

本书针对 ARM920T 核 S3C2410A 芯片,借助 RVDS 和 RealView MDK 集成开发环境,全面详细地介绍了 ARM 的体系结构、硬件接口电路设计以及汇编语言与 C 语言程序设计。全书共分七章,内容包括 ARM 概述与开发环境建设、ARM 体系结构与接口电路、ARM 指令集、ARM 存储系统及其配置、汇编语言程序设计及系统初始化、C 语言程序设计、ARM 接口程序设计。本书的特色在于通过完整的程序实例阐述相关的内容。

本书是作者近几年来从事 ARM 系统设计、开发和教学的结晶。由于 ARM 体系原理的统一性和 ARM C 程序设计的通用性,尽管本书内容偏重于 ARM920T,但是,对 ARM 系列芯片的开发,均具有一定的指导意义。

本书可作为电子技术类、通信类、软件工程类研究生和高年级本科生学习 ARM 芯片级嵌入式程序设计的教材,也可作为从事 ARM 系统开发的电子工程师和软件工程师及嵌入式爱好者的参考书。

★本书作者可以提供书中全部实例的源代码,有需要的读者可联系索取。

### 图书在版编目(CIP)数据

ARM 原理与 C 程序设计/张勇编著. —西安:西安电子科技大学出版社,2009.4

(ARM 嵌入式设计与 Windows CE 开发系列丛书)

ISBN 978-7-5606-2228-6

I. A… II. 张… III. ① 微处理器, ARM ② C 语言—程序设计 IV. TP332 TP312

中国版本图书馆 CIP 数据核字(2009)第 040144 号

策 划 李惠萍

责任编辑 李惠萍

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 西安文化彩印厂

版 次 2009 年 4 月第 1 版 2009 年 4 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 21.875

字 数 519 千字

印 数 1~4000 册

定 价 36.00 元

ISBN 978 - 7 - 5606 - 2228 - 6/TP • 1136

**XDUP 2520001-1**

\*\*\*如有印装问题可调换\*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

## 前 言

---

ARM(Advanced RISC Machine, 高级精简指令集机器)是 ARM 公司设计的 32 位总线高性能微处理器。自 1985 年第一个 ARM1 原型诞生至今, ARM 公司设计的成熟 ARM 体系结构(或称指令集体系结构 ISA)有 ARMv4、ARMv4T、ARMv5TE、ARMv5TEJ、ARMv6 和 ARMv7 等, 对应的处理器家族有 ARM7、ARM9、ARM9E、ARM10E、ARM11、Cortex、SecurCore 和 Xscale 等处理器系列。应用领域涉及商业、军事、航天航空、网络与无线通信、消费电子、医疗电子、仪器仪表和汽车电子等各行各业。

ARM 芯片是封装了一种或多种 ARM 核心以及丰富片上外设的集成块。例如, S3C2410A 芯片集成了 ARM920T 核和包括 USB、串口、IIS 等在内的多个片上外设。ARM 芯片本质上是处理能力和寻址能力更强的单片机。由此, ARM 系统设计包括嵌入了 ARM 芯片的硬件板卡设计和基于 ARM 芯片的软件开发, 即 ARM 原理性设计和程序设计两个部分。其中, ARM 原理性设计在本书中第一至五章介绍, 第六、七两章则主要介绍程序设计。

本书的编排是这样的: 第一章讨论了 ARM 发展历程和应用领域以及 RVDS 仿真开发环境; 第二章阐述了 ARM 体系结构, 包括 ARM 微处理器工作模式、ARM 寄存器、ARM 异常等, 然后, 结合 UP-NETARM2410 实验箱讨论了 S3C2410A 的原理性接口设计; 第三章介绍了 ARM 指令集及寻址方式, 并给出了 ARM 指令的测试程序; 第四章讲述了 Keil RealView MDK 开发环境和 ARM 存储系统及其存储器配置方法; 第五章主要讲述 ARM 系统初始化工作, 由于初始化代码必须由汇编语言编写, 因此, 也全面讲述了汇编语言程序设计方法; 第六章为 C 语言程序设计, 深入介绍了混合语言程序设计和中断程序设计方法; 第七章在讲述 LED 显示和串口通信时, 进一步讲述了中断程序设计方法, 同时, 介绍了 K9F1208 NAND 型 FLASH 访问方法, 并阐述了 S3C2410A 的 BootLoader 方法。

本书的第一至三章使用 RVDS 开发环境, 而第四至七章使用了 RealView MDK 开发环境, 考虑到知识产权、底层编译器和链接器等因素, 这两个软件是统一的。

虽然本书自成体系, 但是没有基础的读者在自学过程中至少需要参考以下技术资料: S3C2410A 芯片资料、ARM920T 体系结构方面的资料([www.arm.com](http://www.arm.com))、K9F1208 芯片资料、ADS 开发导读资料、ARM 指令集方面的资料等, 大部分资料在书中适当位置提示了读者。

本书作为教材使用时,理论课时宜在 30 学时左右,实验课时不能低于 20 学时。

本书中的所有实例都是完整的,但是,读者仍然可以通过 E-mail:zhnyong@21cn.com 和 QQ: 493815991 和作者联系索取源代码。作为一名从事 ARM 相关方面开发和教学的老师,作者将一直跟踪本书的使用情况,并会不断充实和修订本书。由于作者水平有限,敬请读者和同行专家批评指正。

本书与已出版的《Windows CE 应用程序设计》一书(西安电子科技大学出版社出版)同属于 ARM 嵌入式设计与 Windows CE 开发系列丛书。

书中引用的相关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。禁止任何单位或个人摘录本书内容出版发行,严禁依据本书出版实验教程!

张 勇

己丑年于江西财经大学枫林园

# 目 录

|                                  |    |                                    |    |
|----------------------------------|----|------------------------------------|----|
| 第一章 ARM 概述与开发环境 .....            | 1  | 2.4.2 映射存储空间 .....                 | 30 |
| 1.1 嵌入式系统的概念 .....               | 1  | 2.4.3 NOR 型 FLASH 接口电路设计 .....     | 35 |
| 1.1.1 嵌入式系统与 ARM 的关系 .....       | 1  | 2.4.4 NAND 型 FLASH 接口电路设计 .....    | 38 |
| 1.1.2 嵌入式系统与嵌入式操作系统的<br>关系 ..... | 2  | 2.4.5 SDRAM 接口电路设计 .....           | 39 |
| 1.1.3 嵌入式系统的研发特点 .....           | 2  | 2.5 S3C2410 外设接口电路设计 .....         | 42 |
| 1.2 ARM 发展历程及应用领域 .....          | 3  | 2.5.1 片上外设及其映射寄存器 .....            | 42 |
| 1.2.1 ARM 发展历程及命名规则 .....        | 3  | 2.5.2 LED 接口电路设计 .....             | 44 |
| 1.2.2 ARM 微处理器系列 .....           | 4  | 2.5.3 LCD 接口电路设计 .....             | 48 |
| 1.2.3 ARM 微处理器应用领域 .....         | 9  | 2.5.4 触摸屏接口电路设计 .....              | 50 |
| 1.3 ARM 芯片级开发环境 .....            | 9  | 2.5.5 串口接口电路设计 .....               | 51 |
| 1.3.1 仿真环境建设 .....               | 10 | 2.5.6 USB 接口电路设计 .....             | 52 |
| 1.3.2 程序仿真实例与解释 .....            | 13 | 2.5.7 SD 卡接口电路设计 .....             | 53 |
| 本章小结 .....                       | 17 | 2.5.8 鼠标和键盘接口电路设计 .....            | 53 |
|                                  |    | 本章小结 .....                         | 54 |
| 第二章 ARM 体系结构与接口电路 .....          | 18 | 第三章 ARM 指令表 .....                  | 55 |
| 2.1 ARM 微处理器工作模式 .....           | 18 | 3.1 ARM 指令学习程序框架 .....             | 55 |
| 2.2 ARM 寄存器 .....                | 19 | 3.2 ARM 指令格式 .....                 | 56 |
| 2.2.1 通用寄存器 .....                | 20 | 3.3 ARM 指令寻址方式 .....               | 58 |
| 2.2.2 程序状态寄存器 .....              | 21 | 3.3.1 数据处理操作数寻址 .....              | 58 |
| 2.2.3 PC 与指令流水线 .....            | 23 | 3.3.2 字与无符号字节存储/装入寻址 .....         | 61 |
| 2.3 ARM 异常 .....                 | 23 | 3.3.3 双字与半字及有符号字节<br>存储/装入寻址 ..... | 65 |
| 2.3.1 复位异常 .....                 | 25 | 3.3.4 寄存器批量存储/装入寻址 .....           | 67 |
| 2.3.2 中断请求异常 .....               | 25 | 3.3.5 协处理器存储/装入寻址 .....            | 70 |
| 2.3.3 快速中断请求异常 .....             | 26 | 3.4 ARM 指令汇总 .....                 | 72 |
| 2.3.4 软件中断异常 .....               | 27 | 3.4.1 赋值指令 .....                   | 72 |
| 2.3.5 数据中止异常 .....               | 27 | 3.4.2 算术运算指令 .....                 | 73 |
| 2.3.6 预取指令异常 .....               | 27 | 3.4.3 跳转指令 .....                   | 80 |
| 2.3.7 未定义指令异常 .....              | 28 | 3.4.4 比较指令 .....                   | 81 |
| 2.3.8 异常返回地址 .....               | 28 | 3.4.5 逻辑运算指令 .....                 | 82 |
| 2.4 S3C2410 存储器配置及接口电路设计 .....   | 29 | 3.4.6 CPSR 访问指令 .....              | 85 |
| 2.4.1 存储格式 .....                 | 29 |                                    |    |

|                                 |     |                              |     |
|---------------------------------|-----|------------------------------|-----|
| 3.4.7 内存访问指令 .....              | 86  | 5.1.3 汇编语言语法 .....           | 150 |
| 3.4.8 软中断指令 .....               | 95  | 5.2 浮点数据处理 .....             | 152 |
| 3.4.9 协处理器指令 .....              | 95  | 5.2.1 计算基础 .....             | 152 |
| 3.4.10 ARM 伪指令 .....            | 97  | 5.2.2 数值定标 .....             | 152 |
| 本章小结 .....                      | 99  | 5.2.3 程序代码及结果分析 .....        | 153 |
| <b>第四章 ARM 存储系统及其配置</b> .....   | 100 | 5.3 系统初始化 .....              | 159 |
| 4.1 RealView MDK 仿真环境建设 .....   | 100 | 5.3.1 启动代码 S3C2410A.s .....  | 159 |
| 4.1.1 开发平台建设 .....              | 100 | 5.3.2 初始化代码 zyboot.s .....   | 172 |
| 4.1.2 第一个 MDK 工程 .....          | 101 | 5.4 中断服务程序 .....             | 185 |
| 4.1.3 硬仿真环境设置 .....             | 104 | 5.4.1 硬件基础 .....             | 185 |
| 4.1.4 编译配置 .....                | 105 | 5.4.2 EINT2 外部中断及其程序设计 ..... | 186 |
| 4.1.5 在线仿真 .....                | 108 | 5.4.3 SWI 中断及其注意事项 .....     | 192 |
| 4.2 Nand Flash 启动配置 .....       | 109 | 本章小结 .....                   | 193 |
| 4.3 SDRAM 配置 .....              | 112 | <b>第六章 C 语言程序设计</b> .....    | 194 |
| 4.3.1 MDK 启动代码配置 SDRAM .....    | 112 | 6.1 C 程序框架实例 .....           | 194 |
| 4.3.2 自定义 SDRAM 配置代码 .....      | 117 | 6.1.1 跑马灯工程设计 .....          | 194 |
| 4.3.3 堆和栈 .....                 | 120 | 6.1.2 跑马灯程序解释 .....          | 199 |
| 4.4 CP15 协处理器 .....             | 125 | 6.1.3 C 程序框架 .....           | 201 |
| 4.4.1 CP15 协处理器寄存器汇总 .....      | 125 | 6.2 SDRAM 重定位技术 .....        | 203 |
| 4.4.2 寄存器 c0 .....              | 126 | 6.3 C 语言语法 .....             | 205 |
| 4.4.3 寄存器 c13 .....             | 129 | 6.3.1 数据类型 .....             | 205 |
| 4.4.4 寄存器 c1 .....              | 130 | 6.3.2 常用运算符 .....            | 207 |
| 4.4.5 寄存器 c2 .....              | 131 | 6.3.3 程序控制语句 .....           | 211 |
| 4.4.6 寄存器 c3 .....              | 131 | 6.3.4 C 语言函数 .....           | 215 |
| 4.4.7 寄存器 c5 .....              | 131 | 6.3.5 演示实例一 .....            | 216 |
| 4.4.8 寄存器 c6 .....              | 131 | 6.3.6 演示实例二 .....            | 219 |
| 4.4.9 寄存器 c8 .....              | 131 | 6.4 混合语言程序设计 .....           | 223 |
| 4.4.10 寄存器 c10 .....            | 132 | 6.4.1 C 函数嵌套汇编语句 .....       | 223 |
| 4.5 MMU 配置 .....                | 132 | 6.4.2 C 函数调用汇编子程序 .....      | 226 |
| 4.5.1 MMU 工作原理 .....            | 132 | 6.5 中断服务程序设计 .....           | 236 |
| 4.5.2 地址转译 .....                | 134 | 6.5.1 异常入口配置 .....           | 237 |
| 本章小结 .....                      | 137 | 6.5.2 SWI 中断 .....           | 239 |
| <b>第五章 汇编语言程序设计及系统初始化</b> ..... | 138 | 6.5.3 看门狗中断 .....            | 244 |
| 5.1 ARM 汇编语言程序框架 .....          | 138 | 本章小结 .....                   | 248 |
| 5.1.1 通用输入输出 C 口配置 .....        | 140 | <b>第七章 ARM 接口程序设计</b> .....  | 249 |
| 5.1.2 工程 ex5_1 的注解及运行情况 .....   | 147 | 7.1 LED 显示控制 .....           | 249 |
|                                 |     | 7.1.1 看门狗定时器秒表 .....         | 249 |

|                             |     |                             |     |
|-----------------------------|-----|-----------------------------|-----|
| 7.1.2 定时器 1 秒表 .....        | 252 | 本章小结 .....                  | 291 |
| 7.2 K9F1208 访问技术 .....      | 256 | 附录一 S3C2410 专用寄存器地址 .....   | 292 |
| 7.2.1 K9F1208U0C.....       | 256 | 附录二 S3C2410 专用寄存器描述 .....   | 300 |
| 7.2.2 K9F1208 读设计 .....     | 258 | 附 2.1 存储器配置寄存器 .....        | 300 |
| 7.2.3 K9F1208 Boot 设计 ..... | 263 | 附 2.2 NAND FLASH 寄存器 .....  | 304 |
| 7.3 串口通信 .....              | 282 | 附录三 S3C2410 芯片管脚 .....      | 306 |
| 7.3.1 串口配置 .....            | 282 | 附录四 启动代码文件 S3C2410A.s ..... | 313 |
| 7.3.2 串口收发程序 .....          | 283 |                             |     |
| 7.3.3 串口中断编程 .....          | 286 |                             |     |

## 第一章 ARM 概述与开发环境

ARM 即 Advanced RISC Machine(高级精简指令集机器),它是 ARM 公司设计的 32 位总线的高性能微处理器。ARM 公司(WWW.ARM.COM)本身不生产芯片,通过转让或出售 ARM 技术给 OEM(原始设备生产商)专业生产商来生产和销售 ARM 芯片给第三方用户。全球有约 200 家大型半导体生产厂商购买了 ARM 知识产权生产具有 ARM 核的芯片,每秒就有约 90 个 ARM 芯片被使用。

### 1.1 嵌入式系统的概念

数字技术和软件技术是嵌入式系统的核心技术,其中,数字技术包括数字信号处理技术和数字化芯片技术,软件技术包括芯片级的程序设计和操作系统级的程序设计技术。电路系统由传统的模拟电子系统演化为以可编程数字化芯片为核心,添加必要的外设接口实现相应功能的嵌入式系统,在三个相互关联又相对独立的技术领域表现突出,即以单片机为核心的嵌入式控制领域、以 DSP(数字信号处理器)或 FPGA(现场可编程门阵列)为核心的嵌入式数字信号处理领域和以 ARM 或 SOC(片内系统芯片)为核心的嵌入式操作系统及其应用领域。一般地,嵌入式系统被理解为一个相对概念,即在硬件上,它是嵌入在更大规模硬件系统中的电路系统。嵌入式系统的本质在于其硬件系统具有灵活的可编程、可再配置软件等特性,即嵌入式系统必须具有自身的软件系统。

#### 1.1.1 嵌入式系统与 ARM 的关系

广义上讲,凡是嵌入到应用系统中的电子系统可以统称为嵌入式系统,即使通用的计算机系统,如果是嵌入在特定的应用系统中时,也可被称为嵌入式系统。例如,在虚拟仪表系统中用于数据采集、分析和显示的嵌入式计算机系统。狭义上讲,嵌入式系统除具有硬件和软件之外,还要求硬件系统具有体积小、重量轻、功耗低、成本低、可靠性高、可升级等特点,要求软件系统具有体积小、可裁剪性、健壮性、专用性、实时性等特点。因此,从狭义上讲,嵌入式系统硬件往往是以 ARM 芯片为核心的硬件平台,嵌入式系统软件为基于芯片级开发的无操作系统汇编或 C 语言实时性软件,或者是基于嵌入式操作系统开发的图形界面应用程序。

而 ARM 是指 ARM 公司设计的基于 RISC 架构的 32 位高性能微处理器,一般采用哈佛总线结构,具有高速指令缓存和数据缓存,指令长度固定且多级流水执行,具有 MMU(存储器管理单元)和 AMBA(高级微处理器总线结构)总线接口等。ARM 芯片除具有 ARM

核心外，通常还具有丰富的外设接口，例如，外扩 RAM(随机访问存储器)和 FLASH 控制器、LCD 控制器、串行接口、SD 接口、USB 接口、I<sup>2</sup>C 和 I<sup>2</sup>S 总线接口等，此外，ARM 芯片还具有低功耗、体积小等特性。ARM 芯片的高性能、多接口特点决定了其比单片机和 DSP 更适合作为嵌入式系统的核心微处理器，从而，ARM 系统几乎成了嵌入式系统的代名词。

### 1.1.2 嵌入式系统与嵌入式操作系统的关系

一般地，嵌入式系统是面向特定的应用和环境，集成了硬件和软件的单板机。嵌入式系统的硬件资源有限，突出地表现在其具有较小容量的 RAM 和 ROM 空间，通过外扩 SD 卡等存储介质扩展存储空间；嵌入式系统的软件，包括嵌入式操作系统软件，都固化在 FLASH 芯片中。因此，嵌入式操作系统软件体积较小，一般在 32 MB 以下。

嵌入式系统的软件分为两种：其一为直接基于 ARM 芯片开发的汇编或 C 语言实时性程序，这时的程序代码负责管理 ARM 片上所有的资源，包括存储空间和片上外设，程序除根据需要设计特定的功能之外，还要编写 ARM 芯片初始化代码和中断向量表，更重要的是，程序在访问嵌入式系统的硬件时，必须充分考虑硬件接口的时序特点；其二是为嵌入式系统定制多任务、实时的嵌入式操作系统，嵌入式操作系统抽象了嵌入式系统的硬件访问方式，通过提供 API(应用程序接口)函数的方式，在嵌入式操作系统基础上设计用户应用程序，只需调用相应的 API 函数即可，使得嵌入式系统的应用程序设计工作更加简单方便。

嵌入式操作系统也具有桌面操作系统的特点，即管理硬件资源、调度软件进程、处理软件中断等。嵌入式操作系统通常包括硬件驱动软件、系统内核、设备驱动接口、文件系统、图形界面等。嵌入式操作系统要求具有实时性、多任务、模块化、可移植性、可定制等特点，流行的嵌入式操作系统有 Windows CE、嵌入式 Linux、μC/OS-II、VxWorks 等。

因此，嵌入式系统可以表示为：

嵌入式系统 = ARM 硬件系统 + 嵌入式操作系统 + 操作系统级应用软件系统  
或者

嵌入式系统 = ARM 硬件系统 + 芯片级应用软件系统

### 1.1.3 嵌入式系统的研发特点

嵌入式系统研发需要具备电子类和软件类两方面的专门知识，是一门交叉组合型学科。嵌入式系统研发可分为四类，即：

首先，嵌入式系统的硬件平台设计，需要根据应用环境选择合适的 ARM 芯片，满足处理速度和存储深度的要求，同时，需要兼顾性价比和芯片特点与生存周期等因素。ARM 芯片选型后，根据需要的功能添加相应的外设接口处理芯片和电源与时钟芯片等，借助 Protel DXP 等 EDA 软件完成硬件平台的原理性设计和 PCB 设计。目前，嵌入式系统硬件平台的设计基本上实现了模型化设计，即 ARM 芯片与外设芯片的接口电路都形成了规范，只需要按“模型”将 ARM 芯片与所需外设芯片连接起来就可以得到特定的嵌入式系统硬件平台。尽管如此，读懂和分析这个“模型”仍然需要一定的电路基础。

其次，基于 ARM 芯片的芯片级汇编或 C 语言程序设计，这类程序包括系统初始化程

序和特定功能的算法程序,要求设计者对 ARM 芯片工作原理和内部结构有较好的认识和理解,需要对汇编语言和指令以及 C 语言编程有一定的基础。目前,芯片级软件设计达到了框架化的水平,即在现有的框架程序的基础上,添加特定的软件功能达到程序设计的目的。因此,程序员需对框架程序有全面深入的了解,程序员的主要工作集中在使用 C 语言开发算法上。

然后,嵌入式操作系统的定制和驱动程序的开发,这类研发已经完全商业化。设计者可以根据自己选用的 ARM 芯片,直接购买到特定的相兼容的嵌入式操作系统软件,只需要点几下鼠标就可以定制出功能强大的专用嵌入式操作系统,并且,嵌入式操作系统供应商也会提供几乎所有常见外设的驱动程序,例如,触摸屏、LCD 屏、网口、串口、USB 口等。如果设计者想自行研发具有独立知识产权的嵌入式操作系统,那么认真学好开源的  $\mu\text{C}/\text{OS-II}$  嵌入式操作系统是一个建设性的忠告。

最后,基于嵌入式操作系统开发用户应用程序,特别是开发具有良好图形界面的用户应用程序。这是对设计者的一个挑战。基于不同的嵌入式操作系统,开发应用程序的方式有很大的不同。嵌入式 Linux 和 Windows CE 都提供了良好的界面设计支持,分别可以借助 QT 和 Visual Studio 进行应用软件开发。用户可能需要对基于事件消息驱动的编程有进一步的了解,同时,如果是基于 Windows CE,掌握 .Net Framework 编程是一条简捷的路径。2008 年西安电子科技大学出版社出版的《Windows CE 应用程序设计》一书在嵌入式应用程序开发方面能起到较好的引导作用。

本书的中心任务是讨论 ARM 芯片级的程序设计方法,程序设计语言为汇编语言和 C 语言,重点在于讲解 C 语言。本系列丛书中的《Windows CE 系统设计》和《Windows CE .Net Framework 程序设计》(待出版)对嵌入式操作系统的开发、外设驱动及外设应用程序开发进行了详细的阐述。

## 1.2 ARM 发展历程及应用领域

自 1985 年第一个 ARM1 原型诞生至今(ARM 公司成立于 1990 年),ARM 公司设计的成熟 ARM 体系结构(或称指令集体系结构 ISA)有 ARMv4、ARMv4T、ARMv5TE、ARMv5TEJ、ARMv6 和 ARMv7 等,并且版本号还在不断升级,对应的处理器家族有 ARM7、ARM9、ARM9E、ARM10E、ARM11、Cortex、SecurCore 和 XScale 等处理器系列。应用领域涉及商业、军事、航天航空、网络与无线通信、消费电子、医疗电子、仪器仪表和汽车电子等各行各业。

### 1.2.1 ARM 发展历程及命名规则

每个 ARM 处理器都对应于一个特定的 ARM 指令集体系结构版本,例如,ARM920T 微处理器支持指令集体系结构 ARMv4T。ARM 体系结构的发展历程如表 1-1 所示。

表 1-1 中列出了一些典型的 ARM 微处理器名称,ARM 微处理器是根据其具有的功能在“ARM”后添加字母来命名的,目前,这些字母有 T、D、M、I、E、J、F、S,依次表示支持 Thumb 指令集、支持在线 JTAG 调试、内嵌乘法器、支持嵌入式 ICE(在线断点和调

试)技术、增强 DSP 指令、支持 Java 技术、支持向量浮点处理、可综合。ARM 微处理器名称中的数字用于反映处理器系列、存储管理单元以及高速缓存等信息。

表 1-1 ARM 体系结构发展历程

| 版 本                                    | 典型微处理器类型                                   | 特 点                                                                    |
|----------------------------------------|--------------------------------------------|------------------------------------------------------------------------|
| ARMv1 至 ARMv4<br>(1990 年 ARM 公司<br>成立) | 已退市                                        | 早期的版本中只有 ARMv4 在目前某些 ARM7 和 StrongARM 处理器中可见,可以被视为 32 位寻址的 32 位指令集体系结构 |
| ARMv4T<br>(1995 年)                     | ARM7TDMI、ARM7TDMI-S、<br>ARM920T、ARM922T    | 支持 16 位的 Thumb 指令集,比 32 位的 ARM 指令集节省约 35% 的存储空间                        |
| ARMv5TE<br>(1999 年)                    | ARM946E-S、ARM966E-S、<br>ARM968E-S、ARM996HS | 增加了 ARM 与 Thumb 状态切换的指令,增强了 DSP 类型指令,尤其是在语音数字信号处理方面提高了 70% 以上的性能       |
| ARMv5TEJ<br>(2000 年)                   | ARM7EJ-S、ARM926EJ-S、<br>ARM1026EJ-S        | 添加了 Java 加速技术                                                          |
| ARMv6<br>(2001 年)                      | ARM1176JZ(F)-S                             | 改进了异常处理,更好地支持多处理器指令,增加了支持 SIMD(单指令多数据)的多媒体指令,对视频和音频解码性能提高近 4 倍         |
| ARMv6T2                                | ARM1156T2(F)-S                             | 支持 Thumb-2 技术                                                          |
| ARMv7                                  | Cortex-A8、Cortex-A9、<br>Cortex-R4(F)       | 支持 NEON 技术,使得 DSP 和多媒体处理性能提高 4 倍,支持向量浮点运算,为下一代 3D 图像和游戏硬件服务            |
| ARMv7-M                                | Cortex-M3                                  | 优化了微控制器,低功耗                                                            |

1.2.2 ARM 微处理器系列

目前 ARM 微处理器主要有八个系列,即 ARM7 系列、ARM9 系列、ARM9E 系列、ARM10E 系列、ARM11 系列、Cortex 系列、SecurCore 系列、XScale 系列等。各种系列微处理器均有其特点和应用场合。

1. ARM7 微处理器系列

ARM7 微处理器系列内核基于冯·诺依曼体系结构,数据和指令共用相同的总线,内核指令三级流水,支持 ARMv4T 指令集,包括 ARM7TDMI、ARM7TDMI-S、ARM7EJ-S 和 ARM720T 等微处理器核(其中,ARM7EJ-S 执行 ARMv5TEJ 指令集,五级流水,带 Java 加速,可综合;而 ARM720T 已被 ARM926EJ-S 替代)。

ARM7TDMI 是目前非常流行的 32 位微处理器内核,例如, Samsung 公司的 S3C4510B 芯片采用了该内核,它支持 16 位的 Thumb 指令集、快速乘法指令和嵌入式 ICE 调试技术。其 S 变种 ARM7TDMI-S 是可综合的。

ARM720T 处理器核集成了一个 MMU(存储器管理单元)和一个 8 KB 的高速缓存,支持 Windows CE、Linux、Symbian 等实时嵌入式操作系统。

ARM7 系列微处理器主要应用于无线接入手持设备、打印机、数码相机和随身听等。

ARM7 系列微处理器采用 0.13  $\mu\text{m}$ 、0.18  $\mu\text{m}$  或 0.25  $\mu\text{m}$  工艺,主频最高达 130 MIPS,功耗很低,代码与 ARM9、ARM9E 和 ARM10 以及 XScale 处理器兼容。

ARM7TDMI 内核的结构框图如图 1-1 所示。

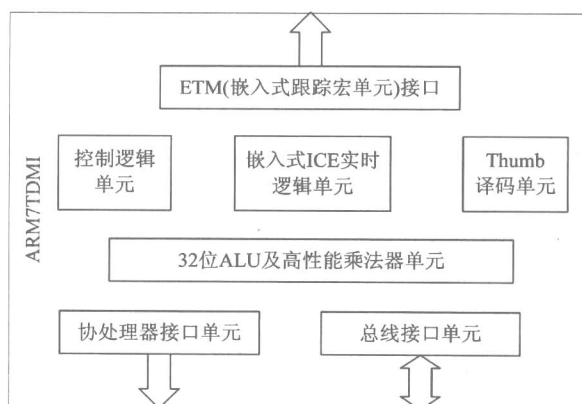


图 1-1 ARM7TDMI 内核结构

## 2. ARM9 微处理器系列

ARM9 及其后更高系列的微处理器核均采用哈佛体系结构，数据总线与指令总线相互独立，数据空间与程序空间相互独立。ARM9 系列微处理器核包括 ARM920T 和 ARM922T 两种，具有五级指令流水线(处理速率可达 1.1 MIPS/MHz)，ARM922T 是 ARM920T 的变种，其数据和指令高速缓存均为 8 KB，而 ARM920T 的数据和指令高速缓存均为 16 KB。

32 位的 ARM9 系列微处理器执行 ARMv4T 指令集，具有两种工作状态，即 Thumb 状态和 ARM 状态，支持 16 位的 Thumb 指令集和 32 位的 ARM 指令集。主频可达 300 MIPS 以上，具有一个 32 位的 AMBA 总线接口，具有 MMU 单元，支持 Windows CE、Linux 和 Symbian OS 等嵌入式操作系统。ARM9 核具有 8 出口的写缓冲器，用于提高对外部存储空间的写速度。ARM9 系列的生产工艺为 0.13  $\mu\text{m}$ 、0.15  $\mu\text{m}$  或 0.18  $\mu\text{m}$ 。

ARM9 系列微处理器可用于 PDA 等高档手持设备、MP5 播放器等数字终端、数码相机等图像处理设备以及汽车电子方面。

ARM920T 核的结构框图如图 1-2 所示。图中，AMBA(Advanced Microprocessor Bus Architecture)为高级微处理器总线结构，AHB(Advanced High-performance Bus)为先进高性能总线，该总线与 APB(Advanced Peripheral Bus)总线协议隶属于 AMBA v2.0 版本。

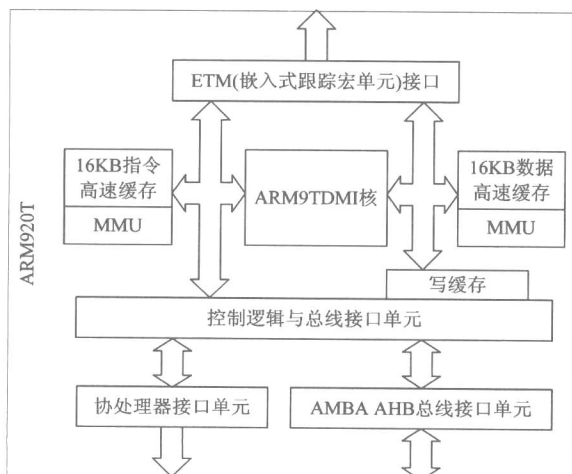


图 1-2 ARM920T 内核结构

ARM920T 核是本书的重点。

### 3. ARM9E 微处理器系列

ARM9E 微处理器系列目前包括 ARM926EJ-S、ARM946E-S、ARM966E-S 和 ARM996HS 等四个种类，都是可综合的，采用五级指令流水技术(速率可达 1.1 MIPS/MHz)，在 0.13  $\mu\text{m}$  工艺下主频可达 300 MIPS，支持 ARM、Thumb 和 DSP 指令集，提供了浮点运算协处理器，可用于图像和视频处理。其中，ARM926EJ-S 微处理器包含了 Jazelle 技术(硬件运行 Java 代码，提高速度近 8 倍)，集成了 MMU，支持 Windows CE、Linux 等嵌入式操作系统。

ARM9E 核具有 16 出口的写缓冲器，用于提高处理器向外部存储空间写数据的速度，支持 ETM9，即具有实时跟踪能力的嵌入式跟踪宏单元，采用软核技术，工艺为 0.13  $\mu\text{m}$ 、0.15  $\mu\text{m}$  或 0.18  $\mu\text{m}$ 。

ARM9E 微处理器可应用于网络通信设备、移动通信设备、图像终端、海量数据存储设备、汽车智能化设备等。

其中，ARM926EJ-S 微处理器执行 ARMv5TEJ 指令集，其内核结构框图如图 1-3 所示，图中，TCM(Tightly Coupled Memory)为紧耦合存储器。

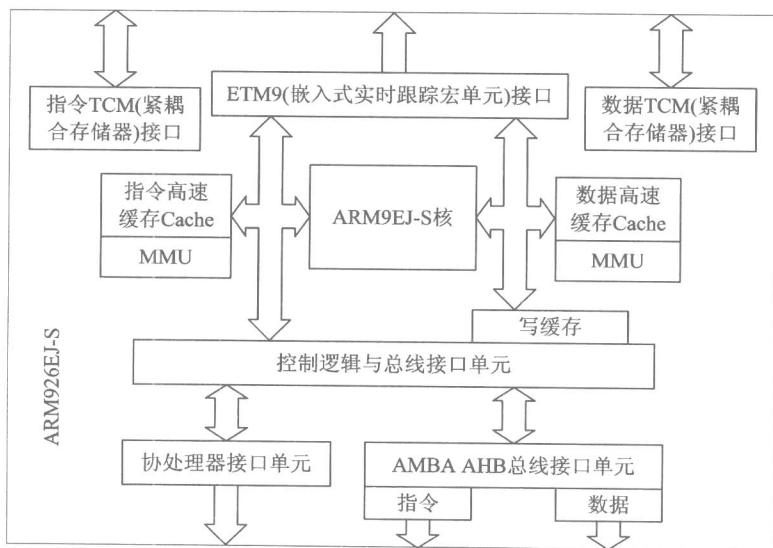


图 1-3 ARM926EJ-S 内核结构

这里解释一下 MMU、TCM 和 Cache 的含义。

MMU(Memory Management Unit)存储器管理单元是 MPU(Memory Protection Unit)存储器保护单元的升级。MPU 将物理存储空间映射到不同的区域，通过设置区域的属性对区域进行访问限制和保护，例如，ARM946E-S 核包含 MPU。此时，ARM 的资源，即存储器系统和外设都映射到某个或某些区域中。MPU 中映射的区域地址与物理地址是一一映射关系，即地址重叠的两个程序将会竞争资源；而 MMU 则是通过页表转换器技术，将实际的物理存储空间映射为虚拟存储器，虚存是独立于物理存储空间的存储空间，允许不同的程序使用相同的虚拟地址(MMU 映射它们到不同的物理地址上)，使得各个程序在各自独立的存储空间中运行而互不影响。

Cache 即高速缓存,位于内核与存储器之间,对于冯·诺依曼体系结构来说,指令和数据共用一个 Cache;对于哈佛体系结构来说,指令 Cache 和数据 Cache 是分开的。由于内核的处理速率一般远远高于总线访问存储器的速率,为了保证内核全速运行,参与处理的数据或指令集一方面先由存储器读到 Cache 备用,另一方面,需要写到存储器的运算结果数据集可由高速 Cache 暂存,因此,Cache 提高了内核的处理速率(高档的处理器又分出一级 Cache、二级 Cache 等)。

Cache 的确提高了内核的性能,但是,程序代码执行的时间变得不可预测,因为,Cache 装载和存储指令和数据的时间不可预测。而 TCM 紧耦合存储器是紧贴内核的高速 SRAM,用于保证取指令或数据操作的准确时钟数,对于要求确定行为的实时算法研究很有帮助。

#### 4. ARM10E 微处理器系列

ARM10E 微处理器系列中主推 ARM1026EJ-S 核,该高性能微处理器核是完全可综合的软核,执行 ARMv5TEJ 指令集,六级指令流水(速度可达 1.35 MIPS/MHz,经 Dhrystone v2.1 测试),支持 ARM、Thumb、DSP 和 Java 指令,支持高性能硬件 Java 字节代码执行,同时具有 MPU 和 MMU,支持实时操作系统和 Windows CE、Linux、Java OS 等嵌入式操作系统。(Dhrystone 是 1984 年 Reinhold P. Weicker 开发的用于测量微处理器运算能力的基准程序,常用于处理器整形运算性能的测量,用 C、Pascal 或 Java 编写,其计量单位为多少次 Dhrystone,后来把在 VAX-11/780 机器上的测试结果 1757 Dhrystones/s 定义为 1 Dhrystone MIPS,即 1 DMIPS。)

ARM1026EJ-S 具有独立的指令高速缓存和数据高速缓存,缓存大小从 4 KB 至 128 KB 可配置;具有独立的数据 TCM 和指令 TCM,TCM 支持插入等待状态,并且大小从 0 至 1 MB 可配置;具有双 64 位/32 位 AMBA AHB 总线接口。ARM1026EJ-S 主要应用于高级手持通信终端、数字消费电子、汽车自动驾驶和复杂工业控制系统等。

ARM1026EJ-S 的内部结构框图如图 1-4 所示,图中的 VIC(Vectored Interrupt Controller)为向量中断控制器。

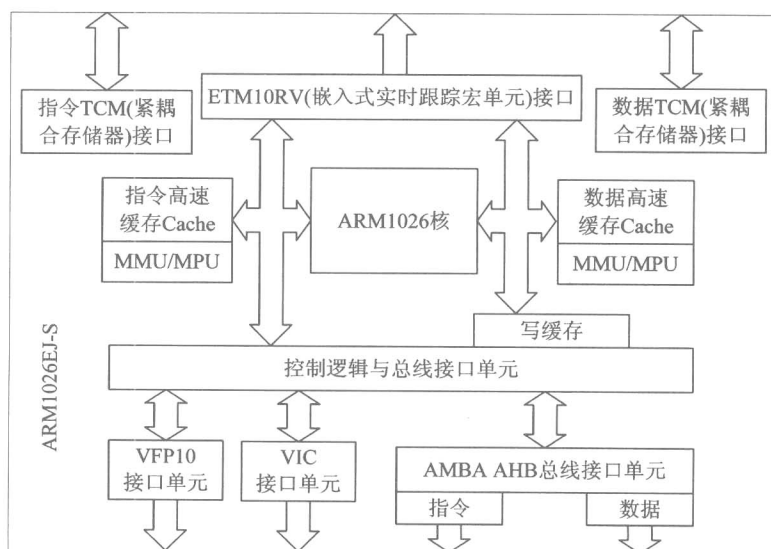


图 1-4 ARM1026EJ-S 内核结构

## 5. ARM11 微处理器系列

ARM11 执行 ARMv6 指令集,指令八级流水执行,于 2003 年发布,包括 ARM1136J(F)-S、ARM1156T2(F)-S、ARM1176JZ(F)-S 单核微处理器和 ARM11 MPCore 多核微处理器(最多四核)等四个系列。ARM11 系列微处理器具有低功耗(0.6 mW/MHz, 0.13  $\mu\text{m}$ , 1.2 V)、处理高性能(例如具有独立的装入/存储和算术运算流水线)、存储高效能(例如具有优化的 TCM 等)等特点,主要应用于数字 TV、机顶盒、游戏终端、汽车娱乐电子、网络设备等。

ARM1136JF-S 的结构框图如图 1-5 所示。

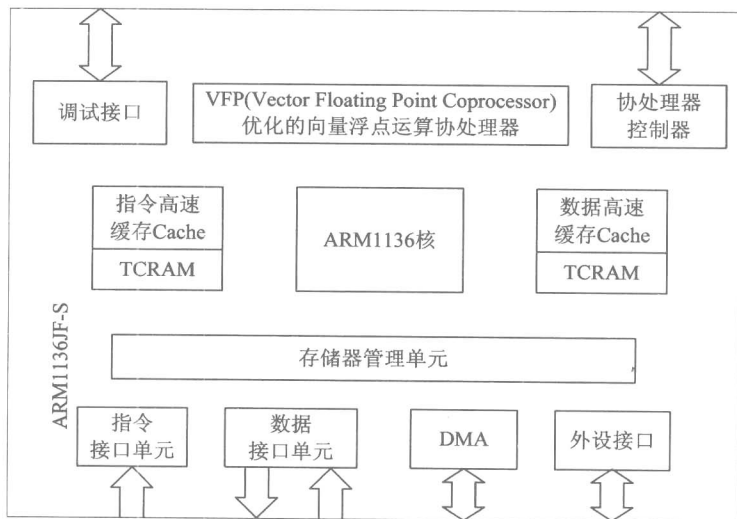


图 1-5 ARM1136JF-S 内核结构

## 6. Cortex 微处理器系列

Cortex 微处理器系列包括三个系列,即 Cortex-A、Cortex-R 和 Cortex-M,均支持 Thumb-2 指令集。其中, Cortex-A 支持复杂操作系统和用户应用,有 Cortex-A8 和 Cortex-A9(单核/多核)等; Cortex-R 面向实时应用,有 Cortex-R4(F)和 Cortex-R4X 等; Cortex-M 进行了内存和功耗优化,仅支持 Thumb-2 指令集,包括 Cortex-M3 和 Cortex-M1 等。

## 7. SecurCore 微处理器系列

SecurCore 系列微处理器面向智能卡、电子商务、银行、身份识别、电子购物等信息安全设备等方面的应用,包括 SC100、SC200 和 SC300 微处理器,具有高性能和极低功耗等特点。其中, SC100 是基于 ARM7TDMI 内核带有 MPU 的安全内核,而 SC200 还支持 Java Card 2.x 加速和其他增强性能。

## 8. XScale 微处理器系列

XScale 微处理器是 StrongARM 的优化改良,独家许可给 Intel 公司(现在 XScale 代工完全转让给 Marvell 公司),基于哈佛结构,具有独立的 32 KB 数据 Cache 和 32 KB 指令 Cache,五级流水,指行 ARMv5TE 架构指令,包括 MMU,具有动态电源管理特性,工作频率可达 1 GHz, 0.18  $\mu\text{m}$  生产工艺,多媒体处理能力得到增强。XScale 微处理器代表芯片为 PXA270 和 PXA320 等,主要应用于 GPS 定位系统、无线网络设备、娱乐和消费电子等。

### 1.2.3 ARM 微处理器应用领域

ARM 微处理器在数据密集型应用(例如视频、图像和数字信号处理等)以及控制密集型应用(例如流程控制、工业控制等)方面均得到了广泛的应用,且具有加载嵌入式操作系统和实时操作系统的能力,因此,ARM 系统在完成特定功能的同时,往往具有友好的人机交互界面,有取代传统的单片机和 DSP 的趋势。

总体而言,ARM 在以下几个方面具有优势:

其一,ARM 芯片的生产与设计是分离的。ARM 公司仅设计 ARM 核,通过出售 ARM 核知识产权给 OEM 公司而与 OEM 公司建立合作关系,OEM 公司可以在 ARM 核的基础上(不能改变 ARM 核)添加特定的外设,生产出具有各自特色的芯片,OEM 公司出售芯片给第三方用户。这种经营运作方式带有全球性、共享性和非垄断性,在 ARM 生产与销售上达到了共享和私有的统一。

其二,ARM 公司推广软核设计。这是一种可定制内核的构架内核技术,面向特定的应用,使得构架后的 ARM 核更具有专用性,而 ARM 内核的构架设计具有通用性,在 ARM 内核设计上达到了专用性与通用性的统一。

其三,ARM 公司推广定制设计。ARM 公司根据第三方用户的需要进行内核定制,要求第二方 OEM 公司进行代工,这种针对第三方用户的定制设计使得 ARM 芯片的应用不但具有专一性,而且能高效地节省成本,即直接针对应用对内核进行了优化和裁剪,同时片上外设进行了相应的去冗留精。在这方面,ARM 芯片达到了应用与设计的统一。

其四,ARM 公司推广 SOC 芯片,即集成了一片或多片 ARM、DSP、FPGA 等数字化芯片的统一内核,用以弥补单核应用的不足。多核处理器主要是面向高端应用,这样,ARM 公司形成了自低端至高端的完整研发策略,且低端至高端的应用具有共同性,每个设计人员从第一次接触 ARM 芯片后,都能在较短的时间内借助“惯性”充分地掌握如何利用 ARM 系列芯片进行特定项目的设计开发。

ARM 的这些特点,使得数字化电子设计的硬件设计和软件开发逐渐走向规范化、标准化和系列化,这对于时间有限的研发人员来说,是期待已久的。研发人员只需要具有一套仿真设备、一套 EDA 软件、一系列 ARM 平台,就可以应对整个数字化领域的研发设计。高等院校是推广 ARM 应用的主要场所,目前几乎所有高校的电子、通信、计算机、软件、应用数学等相关专业都开设了 ARM 类课程。而 ARM 在数字图像处理、数字信号处理、人工智能、机器人、生物医学、特征识别、网络通信、视频处理与压缩、语音处理、雷达技术、编码技术等技术领域都深深涉足。

## 1.3 ARM 芯片级开发环境

ARM 芯片级的开发环境由 ARM 开发板(或实验箱)、仿真器、计算机以及 ARM 开发软件等组成。借助于 ARM 仿真器在线调试程序的过程称为仿真阶段,而将程序下载到 ARM 开发板的 FLASH 上脱离仿真器工作的过程称为自举运行阶段。仿真阶段是自举运行阶段的仿真测试运行准备时期,对项目的整个研发过程起着重要的试金石作用。