

ARM Cortex-M3 内核微控制器初学之路 —— 动手系列

# ARM Cortex-M3

## 内核微控制器

### 快速入门与应用



刘同法 肖志刚 彭继卫 编著

ARM Cortex-M3 内核微控制器初学之路——动手系列

# ARM Cortex-M3 内核 微控制器快速入门与应用

刘同法 肖志刚 彭继卫 编著

北京航空航天大学出版社

## 内 容 简 介

Cortex-M3 是 ARM 公司基于 ARM V7 架构的新型芯片内核。本书首先叙述 Cortex-M3 内核微控制器的内部结构和内部寄存映射及功能,然后通过课题的形式训练读者掌握其编程应用方法。

全书分基础篇和实战篇两部分。基础篇主要讲述 Cortex-M3 的由来和 Cortex-M3 内核微控制器的内部结构。实战篇又分基础训练和应用训练两部分:基础训练主要训练读者对 Cortex-M3 内核微控制器的输入/输出和内部定时器等的应用;应用训练主要训练读者对 Cortex-M3 内核微控制器与外围接口电路进行通信控制的编程方法。

本书既可作为学习 32 位微控制器的单片机爱好者和从事自动控制、智能仪器仪表、电力电子、机电一体化以及各类单片机应用的工程技术人员的学习参考用书,还可作为大学本科、高职高专、技师学院等师生的理论教材或实习教材。

### 图书在版编目(CIP)数据

ARM Cortex-M3 内核微控制器快速入门与应用/刘同法,  
肖志刚,彭继卫编著. —北京:北京航空航天大学出版社,2009.8  
ISBN 978-7-81124-878-4

I. A… II. ①刘…②肖…③彭… III. 微控制器 IV. TP332.3

中国版本图书馆 CIP 数据核字(2009)第 142934 号

© 2009,北京航空航天大学出版社,版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书内容。  
侵权必究。

### ARM Cortex-M3 内核微控制器快速入门与应用

刘同法 肖志刚 彭继卫 编著

责任编辑 冯 颖

\*

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(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:emsbook@gmail.com

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:787×1092 1/16 印张:29.25 字数:749 千字

2009 年 8 月第 1 版 2009 年 8 月第 1 次印刷 印数:4 000 册

ISBN 978-7-81124-878-4 定价:48.00 元

# 前 言

时光飞逝。写完了《单片机 C 语言编程基础与实践》，本想按原计划写一本 C 语言在单片机高层中的应用实践之类的书，来完成未尽事宜（比如单片机的互联工程、74HC595 级联在电子点阵屏中的运用、CAN 总线通信等）。可 ARM 公司的 Cortex-M3 内核的 32 位微控制器已出现在眼前，这使我不得不迅速改变方向，走向新的微控制器世界。

2008 年夏天，正值暑假，我又来到广州拜访我的老师——周立功先生。此时距离我第一次来到这里已是四年过去了，虽然此后每年都来，但却从未到工程部拜访过昔日指导过我的老师们，于是这次多安排了一些时间到我曾经学习过的 ARM 工程小组看看。没想到仅仅是几年的时间，昔日的 ARM 小组已经发生了惊人的变化！2004 年我在这里学习 DP-51 时，他们的 ARM 开发才刚刚起步，2005 年时发展成为一个开发小组，但是现在却已经发展成为 ARM 工程部了！并细分为 Windows CE 小组、Linux 小组、ARM Cortex-M3 内核微控制器团队等，已经达到了几百人的规模，真可谓“兵强马壮，好不热闹”！

原本我并没有打算深入 ARM 的世界，只想利用现有的单片机做点实实在在的事情。但是在听取了 ARM Cortex-M3 内核微控制器团队对 Cortex-M3 核的介绍之后，我却兴奋不已！就这样，我立即作出了新的学习计划，并开始了对 ARM Cortex-M3 内核微控制器在工程中的应用的学习。

学无止境。记得我的老师——周立功先生曾对我说过，企业接触到的新技术远比学校要来得快。

为了让学生和读者更快地接触到新的技术知识，我又决定开始为期 3 个月的新器件新技术的学习。我希望通过我的学习，为大家节约从对新器件的认识应用到开发的时间；通过我的理解，使朋友们能在一到二周的时间内用上一款新的芯片设计工程。《ARM Cortex-M3 内核微控制器快速入门与应用》就是这样一本书。

本书在编排上紧紧围绕微控制器的共性，即 I/O、定时器、中断系统、UART 通信等，一步步地回答初学者提出的问题。譬如，新芯片的输入/输出在哪里？如何编程控制？“请您看课题 1 和课题 2，在那里我对 Cortex-M3 内核微控制器的输入/输出（GPIO）作了详尽讲解并给出了实例！”。又如，我们在单片机中常用到定时器，那么 Cortex-M3 内核微控制器中有定时器吗？“有！Cortex-M3 内核微控制器带有两个通用定时器，请您读课题 3”。再如，在单片机中我们常常用到中断控制、UART 通信，其在 Cortex-M3 内核微控制器中是如何编程应用的呢？“请您看看课题 2 和课题 4，有现成的代码，只要您能用就直接搬去吧！”。全书通过这样的一种构思，将我们所熟悉的单



片机与 Cortex-M3 内核微控制器联系起来。

江水滔滔,流尽的是人的时间。

对于生活在计算机时代的你和我,快速、有效的学习才是积累知识和财富的法宝。

基于上述想法,本书在内容编排结构上分为三部分:其一,基础认识部分分为 5 个章节,重点介绍了 ARM 体系结构的发展过程、Cortex-M3 内核的内部结构以及 Cortex-M3 内核微控制器芯片 LM3S101(102)的内部结构与寄存器映射;其二,Cortex-M3 内核微控制器内部资源训练部分分为 7 个课题,按照 GPIO 口输出、GPIO 口输入、通用定时器的启动与应用、通用 UART 串行通信的启动与应用、同步串行通信口(SSI)的启动与应用、模拟比较器的应用、看门狗的启动与应用的顺序进行介绍;其三,Cortex-M3 内核微控制器外围接口电路应用分为 8 个课题,示范性地讲解了 SPI 通信器件、I<sup>2</sup>C 通信器件在 Cortex-M3 内核微控制器上的应用。我在学习外围接口电路在 Cortex-M3 内核微控制器上的应用的过程中,最大的体会就是时序和 GPIO 引脚控制。掌握好这两个要点,使用时如同单片机对外围接口电路编程一样。

上述编排完全与学习单片机的顺序同步,可以有效地提高读者的学习速度。只要读者认真学习和掌握 Cortex-M3 内核微控制器的编程特征,很快就可以上手到应用。

2008 年即将过去,又值岁末年初之际,在此将小时候读过的一首诗与各位共勉:

劝君莫惜金缕衣,  
劝君须取少年时。  
花开堪折直须折,  
莫待无花空折枝。

.....

读过我的书的读者,你一定会感觉到每学完一本书都是上升了一个台阶吧。为完成我的这一教学思想还需要编写 4 本书,分别是:《C51 单片机学习与应用速成》、《ARM Cortex-M3 内核微控制器外围接口电路与工程实践(一)》、《ARM Cortex-M3 内核微控制器外围接口电路与工程实践(二)》以及《 $\mu$ C/OS-II 实时操作系统在工程中的应用——基于 ARM Cortex-M3 内核微控制器》。

愿本书能为各位读者的学习节约更多宝贵的时间。

本书由刘同法、肖志刚、彭继卫共同编写,眭仁武老师负责审稿。衡阳技师学院电气技师班的张洁、陈瑞龙、王军林、李纳、李奔、周明正、蒋育满、伍要明、李杨勇、汤柯夫、刘聪、樊亮等同学参与了本书的程序测试工作。

感谢周立功先生对本人的大力支持,感谢周立功单片机发展有限公司各位老师的及时解答和指导,感谢博圆周立功单片机 & 嵌入式系统培训部全体人员对本人的大力支持和帮助,感谢南华大学的张翼、李孟雄以及湖北工程学院的江山等同学对本书进行的体验性学习。

由于作者水平有限,书中错误和不妥之处在所难免,恳请读者批评指正。有兴趣的读者可以发送邮件到 bymcupx@126.com 与作者进一步交流;也可以发送邮件到 buaafy@sina.com,与本书策划编辑进行交流。

作者  
2009 年 5 月

# 学习本书的方法

一本书的完成是按一定的教学思想、编写思路写出来的,而不是随随便便拼凑起来的。实际上我们的学习也是一样。毛主席在谈到读书的问题时说,只要精读一本就可以触类旁通了。自学习单片机以来,我一直认为学好单片机的中心问题就是学好 I/O 口,因为数据必须要通过这些引脚传出去,也要通过这些引脚读回来。

对于学过单片机的朋友来说,在学习新的单片机时很自然就会想到新的 I/O 口在哪里,有多少个,如何操控?也就是如何将数据和命令发出,如何将输入的信息读回等。这也是每一个学习单片机的朋友要问的问题。所以在本书的编排上首先将对引脚的 GPIO 口操控放在课题 1 中给出。学习中首先要找到这一块。ARM Cortex-M3 内核微控制器的 GPIO 口(I/O)的最大特性是由外设模块 PORTA、PORTB、PORTC 管理,启动引脚时必须要先设置外设。其他的控制寄存器分别是方向寄存器(DIV)、模式控制寄存器(AFSEL)、数据寄存器(DATA)。只要设置好这些寄存器,各功能就可以实现了。操作时,需要输出信号时将引脚的 DIV 方向设置寄存器设为输出,需要输入信号时将 DIV 设为输入,引脚便可以收到信号。

在学习 ARM Cortex-M3 内核微控制器时我们要明白一件最重要的事,那就是遍地都是寄存器,这和我们在学习 8 位单片机外围接口电路时有些类似。实际上我们在学习 SPI 和 I<sup>2</sup>C 通信时一个明显的感觉就是:SPI 使用的全是命令,只要掌握了 SPI 通信中的命令,便可以灵活地操控此类芯片;而 I<sup>2</sup>C 通信操控的是寄存器,只要能将数据正确地写入芯片的内部寄存器,便可以随心所欲地操控这一块芯片。所以在对 ARM Cortex-M3 内核微控制器芯片操作时,是与 I<sup>2</sup>C 有相似之处的,而且是大面积的寄存器。

其次要重点学习的是,ARM Cortex-M3 内核微控制器内部的定时器/计数器系统。每一个生物都有自己的时钟系统,单片机也不例外。掌握单片机定时系统的启用与停止,这是必不可少的。ARM Cortex-M3 内核微控制器系列中的 LM3S101(102)芯片带有两个 32 位的定时器/计数器。其中每一个定时器都有 A、B 之分,并可以设为 16 位的计数器或定时器。学习中如果这一部分确实看不懂,可以先将示范程序抄用,以后再慢慢弄懂。

再次要重点学习的是,ARM Cortex-M3 内核微控制器内部的中断系统。在学习中断时一定要搞清两件事,一是如何启动中断,二是如何创建中断服务子程序。

从 GPIO 引脚到各模块中断,我发现 ARM Cortex-M3 内核微控制器的函数库中都带有中断申请服务子程序名的函数,如“GPIOPortIntRegister();”为注册一个中断服务函数名。这是一个 GPIO 端口中断注册中断服务函数名的函数。所以在学习中断



处理时,这是最好的方法。这种学习不同于在 8 位单片机中的学习,后者毕竟是小系统。如果在学习中有朋友无法弄懂的话,也可以直接抄用。

掌握了 GPIO 口的输入/输出,定时器/计数器的启用,中断系统的启用,实质上已经掌握了这一类型的芯片,余下的就只有芯片中新增加的部分了。

下面是我学习中的一点体会:

虽然经历了多种单片机的学习过程,但在单片机的编程中使用 API 函数这还是第一次。曾经在对 Windows 操作系统中运用 SDK 库的编程过程中用到过 API 函数,所以这对于我来说是非常熟悉的事情,并且通过这一次的学习使我对 API 函数有了更深一层的理解。过去在编写 Windows 应用程序时没有接触到硬件,现在使用 API 函数是直接对硬件进行编程,让我明白了对寄存器进行设置时必须使用系统先说明好的标识符(宏定义)。当然了,如果你熟悉功能寄存器地址,也可以对功能寄存器进行直接编程。但由于 32 位微控制器的硬件系统非常庞大,所以一下子完全掌握是有一定难度的。为了能够快速、高效地学习并掌握一种新芯片的应用,直接使用公司提供的 API 函数和标识符是最好的选择。不过如果你到了一定的境界,直接操控寄存器那当然更好了。

对于前面我所提到的标识符,还是举例来说明一下吧。比方说,我们在对外接芯片进行通信编程时常常会用到上升沿、下降沿、低电平这样的字眼,那么在对 ARM Cortex-M3 内核微控制器编程时如何表示呢?请看下面的 C 程序宏定义:

```
#define GPIO_FALLING_EDGE    0x00000000    // Interrupt on falling edge    下降沿
#define GPIO_RISING_EDGE     0x00000004    // Interrupt on rising edge     上升沿
#define GPIO_LOW_LEVEL       0x00000002    // Interrupt on low level      低电平
```

在启用中断时就会用到这些标识符。请看:

```
//3. 设置 PA0~PA4_KEY 中断的触发方式为 LOW[低电平]触发
GPIOIntTypeSet(GPIO_PORTA_BASE, PA0TO3_KEYINT, GPIO_LOW_LEVEL);
```

在程序中就是这样使用的,不用考虑寄存器的具体地址。GPIOIntTypeSet() 为 API 函数。

在学习中千万要注意的是,很多功能寄存器都在 API 的函数库中得到了全部的定义,所以学习时直接学会运用这一块就可以了。当然了,如果在学习中能对各功能寄存器有一定的了解和掌握更好,起码在有些功能不能实现时可以直接操作寄存器(对寄存器进行直接赋值)。

其次,熟悉 API 函数名称中单词的意义也是很重要的。用过 API 函数的朋友一看函数名就知道其功能和用途。在运用 C 语言编程的今天,熟悉 API 函数名最大的好处是可使实施程序规范化、标准化。

比方说有一个这样的函数名: IntEnable()。一看就知道 Int 为中断, Enable 为允许或使能。连起来就是中断使能函数,要启用中断非它莫属。

再看一个: IntDisable()。Int 为中断, Disable 为禁止。连起来就是中断禁止函数。

再看一个: GPIOPortIntRegister()。GPIO 为输入/输出引脚, Port 为端口, 这里是指 PA 口或 PB 口, Int 为中断, Register 为注册。所以这个函数其意是注册一个端口的

服务子程序的函数名称,比如现在启用的是 PA 口,就用于为 PA 口注册一个中断服务函数名。

还有更多的单词,比如 Read(读)、Write(写)、Mode(模式)、Config(配置)等。熟记和掌握这些单词对于学习 API 函数的应用是很有帮助的。

C 语言是一种编程十分灵活的语言,怎样写都行。现在各大企业为了更好地实现团队协作,都提出规范化的要求,所以在学习时不要随心所欲地写,一定要学习遵守程序规范。

这是本人在学习中的一点浅薄之见,希望能给您的学习带来一些便利。

万丈高楼平地起,有了好的地基才可以盖高楼。

对于本书的学习,需要良好的 C 语言编程功底(基础),最好还要有一点 8 位单片机基础。切记。

作者  
2009 年 5 月



# 目 录

## 基 础 篇

### 第 1 章 ARM 公司与其体系结构概述

1.1 ARM 公司概况 .....	2
1.1.1 ARM 产品领域 .....	2
1.1.2 ARM 产品特点 .....	3
1.2 ARM 体系结构概述 .....	6
1.2.1 ARM CPU 应用分类 .....	6
1.2.2 应用处理器 .....	6
1.2.3 嵌入式控制处理器 .....	7
1.2.4 ARM 体系结构的发展 .....	7
1.2.5 ARM 技术发展趋势 .....	9

### 第 2 章 ARM Cortex-M3 处理器内核结构

2.1 Cortex-M3 内核 .....	11
2.2 Thumb-2 指令集架构 .....	13
2.3 嵌套向量中断控制器(NVIC) .....	14
2.4 存储器保护单元(MPU) .....	15
2.5 调试和跟踪 .....	16
2.6 总线矩阵和接口 .....	17
2.7 Cortex-M3 指令系统 .....	17

### 第 3 章 Cortex-M3 内核微控制器 LM3S101/LM3S102 硬件结构

3.1 概 述 .....	20
3.2 引脚功能 .....	21
3.2.1 引脚分布 .....	21
3.2.2 引脚功能描述 .....	22
3.3 硬件结构 .....	24
3.4 ARM Cortex-M3 内核 .....	25
3.5 内存储器单元(Flash/SRAM) .....	26



3.5.1	SRAM 存储器	26
3.5.2	Flash 存储器	26
3.6	中断系统	28
3.7	通用输入/输出 (GPIO)	48
3.7.1	GPIO 功能模块	49
3.7.2	数据寄存器操作	49
3.7.3	数据方向	51
3.7.4	中断控制	51
3.7.5	模式控制	51
3.7.6	引脚配置	51
3.7.7	标识 (Identification)	52
3.8	通用定时器	52
3.8.1	硬件模块框图	52
3.8.2	功能描述	52
3.9	看门狗定时器	58
3.9.1	看门狗模块框图	58
3.9.2	功能描述	58
3.10	通用异步串行通信	59
3.10.1	硬件方框图	59
3.10.2	功能描述	59
3.11	同步串行通信接口 (SSI)	62
3.11.1	SSI 模块框图	63
3.11.2	功能描述	63
3.12	I <sup>2</sup> C 接口	71
3.12.1	I <sup>2</sup> C 硬件方框图	71
3.12.2	功能描述	71
3.13	模拟比较器	76
3.13.1	硬件方框图	77
3.13.2	功能描述	77
3.13.3	内部参考编程	78
3.14	JTAG 接口	79
3.14.1	硬件方框图	80
3.14.2	功能描述	80
3.15	系统存储器映射	84
3.16	系统控制	85
3.16.1	功能描述	85
3.16.2	初始化和系统配置	89
3.16.3	系统控制寄存器的映射	90
3.16.4	系统控制寄存器可实现功能描述	91

**第 4 章 对 C 语言的回顾**

4.1 指针的应用 .....	108
4.2 左移、右移和位逻辑符号在程序中的应用 .....	109
4.3 #define 常数定义符 .....	110
4.4 const(常数变量) .....	111
4.5 #if #endif (条件编译) .....	112
4.5.1 条件编译命令的第一种格式 .....	112
4.5.2 条件编译命令的第二种格式 .....	113
4.5.3 条件编译命令的第三种格式 .....	114
4.6 typedef(用户自定义类型) .....	114
4.6.1 基本类型的自定义 .....	114
4.6.2 数组类型的自定义 .....	115
4.6.3 结构型、共用型的自定义 .....	115
4.6.4 指针型的自定义 .....	116

**第 5 章 IAR Embedded Workbench 与 LM LINK JTAG 快速入门**

5.1 IAR Embedded Workbench 的安装和使用 .....	117
5.1.1 IAR Embedded Workbench 的安装 .....	117
5.1.2 安装 Luminary Stellaris 芯片资源文件与 LM LINK JTAG 驱动程序 .....	119
5.1.3 IAR Embedded Workbench 的使用 .....	122
5.2 程序的编译与调试 .....	135

**实 战 篇****第 6 章 Cortex-M3 内核微控制器 LM3S101(102) 内部资源应用实践**

课题 1 LM3S101(102)基本的输入/输出 GPIO 应用练习 .....	138
课题 2 LM3S101(102) GPIO 按键信号输入与中断功能的应用方法 .....	155
课题 3 定时器/计数器(含中断)的启动与运用 .....	183
课题 4 通用 UART 串行通信的启动与应用 .....	221
课题 5 同步串行通信口(SS1)的启动与应用 .....	243
课题 6 LM3S101(102)模拟比较器的应用 .....	264
课题 7 LM3S101(102)看门狗的启动与应用 .....	278

**第 7 章 Cortex-M3 内核微控制器 LM3S101(102) 外围接口电路在工程中的应用**

课题 8 模拟 SPI 通信 FM25L04 存储芯片在 LM3S101(102)系统中的应用 .....	291
课题 9 LCD_JCM12864M 的在 LM3S101(102)单片机上的应用 .....	308



课题 10	模拟 I <sup>2</sup> C 通信在 LM3S101(102) 芯片中的应用(at24Cxx) .....	335
课题 11	用 8 位数码管显示 LM3s101(102) 内部 RTC 实时时钟(ZLG7290 驱动) .....	356
课题 12	LCD_TC1602 在 LM3S101(102) 系统中的应用(74HC595 串并转换) .....	392
课题 13	PCF8563 时钟芯片在 LM3S101(102) 系统中的运用 .....	413
课题 14	步进电机的细分控制在 LM3S101(102) 系统中的运用 .....	430
课题 15	使用 JTAG 引脚作普通的 GPIO 口 .....	443
附录 A	Cortex-M3 内核微控制器 LM3S101(102) 最小系统 .....	452
附录 B	网上资料内容说明 .....	454
参考文献	.....	455
温馨提示	.....	456

# 基础篇

- ARM 公司与其体系结构概述
- ARM Cortex-M3 内核结构
- LM3S101/LM3S102 硬件结构
- 对 C 语言的回顾
- IAR Embedded Workbench 快速入门

# 第 1 章

## ARM 公司与其体系结构概述

### 1.1 ARM 公司概况

ARM(Advanced RISC Machines)既是一个公司的名字,也是对一类微处理器的通称,还可以认为是一种技术的名字。

ARM 公司 1991 年成立于英国剑桥,主要出售芯片设计技术的授权。目前,采用 ARM 技术知识产权(IP)核的微处理器(即我们通常所说的 ARM 微处理器),已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场。基于 ARM 技术的微处理器,其应用占据了 32 位 RISC 微处理器 75% 以上的市场份额。ARM 技术正在逐步渗透到我们生活的各个方面。

#### 1.1.1 ARM 产品领域

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司。作为知识产权供应商,它本身不直接从事芯片生产,而是转让设计许可由合作公司生产各具特色的芯片。世界各大半导体生产商从 ARM 公司购买其设计的 ARM 微处理器核,根据各自不同的应用领域,加入适当的外围电路,从而形成自己的 ARM 微处理器芯片进入市场。目前,全世界有几十家大的半导体公司都使用 ARM 公司的授权,因此既使得 ARM 技术获得了更多的第三方工具、制造、软件的支持,又使得整个系统成本降低,产品更容易进入市场被消费者接受,更具有竞争力。到目前为止,ARM 微处理器及技术的应用几乎已经深入到各个领域。

**工业控制领域:** 作为 32 位的 RISC 架构,基于 ARM 核的微控制器芯片不但占据了高端微控制器市场的大部分市场份额,同时也逐渐向低端微控制器应用领域扩展。ARM 微控制器的低功耗、高性价比,向传统的 8 位/16 位微控制器提出了挑战。

**无线通信领域:** 目前已有超过 85% 的无线通信设备采用了 ARM 技术。ARM 以其高性能和低成本的优势,在该领域的地位日益巩固。

**网络应用:** 随着宽带技术的推广,采用 ARM 技术的 ADSL 芯片正逐步获得竞争优势。此外,ARM 在语音及视频处理上得到了优化,并获得了广泛的支持,也对 DSP 的应用领域提出了挑战。

**消费类电子产品:** ARM 技术在目前流行的数字音频播放器、数字机顶盒和游戏机中得到广泛采用。

**成像和安全产品:** 现在流行的数码相机和打印机中绝大部分采用 ARM 技术。手机中的 32 位 SIM 智能卡也采用了 ARM 技术。

除此以外,ARM 微处理器及技术还应用到其他不同的领域。

### 1.1.2 ARM 产品特点

采用 RISC 架构的 ARM 微处理器一般具有如下特点:

- ▶ 体积小、功耗低、成本低、性能高;
- ▶ 支持 Thumb(16 位)/ARM(32 位)双指令集,能很好地兼容 8 位/16 位器件;
- ▶ 大量使用寄存器,指令执行速度更快;
- ▶ 大多数数据操作都在寄存器中完成;
- ▶ 寻址方式灵活简单,执行效率高;
- ▶ 指令长度固定。

ARM 微处理器目前包括下面几个系列,以及其他厂商基于 ARM 体系结构的处理器。除了具有 ARM 体系结构的共同特点之外,每一个系列的 ARM 微处理器都有各自的特点和应用领域。

- ARM7 系列;
- ARM9 系列;
- ARM9E 系列;
- ARM10E 系列;
- SecurCore 系列;
- Intel 的 XScale;
- Intel 的 StrongARM;
- Cortex 系列。

其中,ARM7、ARM9、ARM9E 和 ARM10E 为 4 个通用处理器系列,每个系列提供一套相对独特的性能来满足不同应用领域的需求,如 SecurCore 系列专门为安全要求较高的应用而设计。Cortex 是 ARM 公司新设计的体系结构,基于 ARM v7 架构的处理器将给系统设计者提供更合适的 CPU 选择。

#### 1. ARM7 系列微处理器

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器,最适合用于对价位和功耗要求较高的消费类应用。ARM7 系列微处理器系列具有如下特点:

- ▶ 具有嵌入式 ICE - RT 逻辑,调试开发方便;
- ▶ 极低的功耗,适合对功耗要求较高的应用,如便携式产品;
- ▶ 能够提供 0.9 MIPS/MHz 的三级流水线结构;
- ▶ 代码密度高并兼容 16 位的 Thumb 指令集;
- ▶ 指令系统与 ARM9 系列、ARM9E 系列、ARM10E 系列兼容,便于用户为产品升级换代。
- ▶ 主频最高可达 130 MIPS,高速的运算处理能力能胜任绝大多数的复杂应用。ARM7 系列微处理器主要用在工业控制、Internet 设备、网络和调制解调器设备、移动电话等多媒体和嵌入式应用领域。

ARM7 系列微处理器包括如下几种类型的内核:ARM7TDMI、ARM7TDMI - S、ARM720T、ARM7EJ。其中,ARM7TDMI 是目前使用最广泛的 32 位嵌入式 RISC 处理器,



属低端 ARM 处理器核。TDMI 的基本含义如下：

- T:支持 16 位压缩指令集 Thumb;
- D: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。

### 2. ARM9 系列微处理器

ARM9 系列微处理器在高性能和低功耗特性方面提供最佳的性能,具有以下特点:

- ▶ 5 级整数流水线,指令执行效率更高;
- ▶ 提供 1.1 MIPS/MHz 的哈佛结构;
- ▶ 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- ▶ 支持 32 位的高速 AMBA 总线接口;
- ▶ 全性能 MMU,支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- ▶ MPU 支持实时操作系统;
- ▶ 支持数据 Cache 和指令 Cache,具有更高的指令和数据处理能力。

ARM9 系列微处理器主要应用于无线设备、仪器仪表、安全系统、机顶盒、高端打印机、数字照相机和数字摄像机等。

ARM9 系列微处理器包含 ARM920T、ARM922T 和 ARM940T 三种类型,以适用于不同的应用场合。

### 3. ARM9E 系列微处理器

ARM9E 系列微处理器为综合处理器,使用单一的处理器内核提供了微控制器、DSP、Java 应用系统的解决方案,极大地减小了芯片的面积和系统的复杂程度。

ARM9E 系列微处理器提供了增强的 DSP 处理能力,很适合于那些需要同时使用 DSP 和微控制器的应用场合。

ARM9E 系列微处理器的主要特点如下:

- ▶ 支持 DSP 指令集,适合于需要高速数字信号处理的场合;
- ▶ 5 级整数流水线,指令执行效率更高;
- ▶ 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- ▶ 支持 32 位的高速 AMBA 总线接口;
- ▶ 支持 VFP9 浮点处理协处理器;
- ▶ 全性能 MMU,支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- ▶ MPU 支持实时操作系统;
- ▶ 支持数据 Cache 和指令 Cache,具有更高的指令和数据处理能力;
- ▶ 主频最高可达 300 MIPS。

ARM9E 系列微处理器主要应用于下一代无线设备、数字消费品、成像设备、工业控制、存储设备以及网络设备等领域。

ARM9E 系列微处理器包含 ARM926EJ-S、ARM946E-S 和 ARM966E-S 三种类型,以适用于不同的应用场合。



#### 4. ARM10E 系列微处理器

ARM10E 系列微处理器具有高性能、低功耗的特点。由于采用了新的体系结构,与同等的 ARM9 器件比较,在同样的时钟频率下,性能提高了近 50%。同时,ARM10E 系列微处理器采用了两种先进的节能方式,使其功耗极低。

ARM10E 系列微处理器的主要特点如下:

- ▶ 支持 DSP 指令集,适合于需要高速数字信号处理的场合;
- ▶ 6 级整数流水线,指令执行效率更高;
- ▶ 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- ▶ 支持 32 位的高速 AMBA 总线接口;
- ▶ 支持 VFP10 浮点处理协处理器;
- ▶ 全性能 MMU,支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- ▶ 支持数据 Cache 和指令 Cache,具有更高的指令和数据处理能力;
- ▶ 主频最高可达 400 MIPS;
- ▶ 内嵌并行读/写操作部件。

ARM10E 系列微处理器主要应用于下一代无线设备、数字消费品、成像设备、工业控制、通信和信息系统等领域。

ARM10E 系列微处理器包含 ARM1020E、ARM1022E 和 ARM1026EJ - S 三种类型,以适用于不同的应用场合。

#### 5. SecurCore 系列微处理器

SecurCore 系列微处理器专为安全需要而设计,提供了完善的 32 位 RISC 技术的安全解决方案,因此 SecurCore 系列微处理器除了具有 ARM 体系结构低功耗、高性能的特点外,还具有其独特的优势——提供了对安全解决方案的支持。

SecurCore 系列微处理器除了具有 ARM 体系结构的主要特点外,还在系统安全方面具有如下特点:

- ▶ 带有灵活的保护单元,以确保操作系统和应用数据的安全;
- ▶ 采用软内核技术,防止外部对其进行扫描探测;
- ▶ 可集成用户自己的安全特性和其他协处理器。

SecurCore 系列微处理器主要应用于一些对安全性要求较高的应用产品及应用系统中,如电子商务、电子政务、电子银行业务、网络和认证系统等。

SecurCore 系列微处理器包含 SecurCoreSC100、SecurCoreSC110、SecurCoreSC200 和 SecurCoreSC210 四种类型,以适用于不同的应用场合。

#### 6. XScale 处理器

XScale 处理器是基于 ARMV5TE 体系结构的解决方案,是一款全性能、高性价比、低功耗的处理器。它支持 16 位的 Thumb 指令和 DSP 指令集,用于数字移动电话、个人数字助理和网络产品等场合。

XScale 处理器原由 Intel 公司设计推出,但目前该技术已经出售给 Marvell 公司。

Intel Strong ARM SA - 1100 处理器采用 ARM 体系结构高度集成的 32 位 RISC 微处理器。它融合了 Intel 公司的设计和处理技术以及 ARM 体系结构的电源效率,采用在软件上兼