



# Design and Practice of Embedded System on ARM

# ARM 嵌入式系统设计及实践

· 杨 恒 主编



西安电子科技大学出版社

<http://www.xduph.com>

# ARM 嵌入式系统设计及实践

杨 恒 主编

西安电子科技大学出版社

2005

## 内 容 简 介

嵌入式技术是微电子技术与计算机技术中的一个重要分支。本书系统地介绍了有关 ARM 嵌入式系统的发展历程、指令集及其操作系统,并以 Philips 公司的 ARM7TDMI 器件 LPC2210 为例,结合延时控制蜂鸣器、LED 显示、LCD 液晶显示、以太网接口以及基于实时操作系统  $\mu\text{C}/\text{OS-II}$  的实时时钟设计等,由浅入深地详述了应用 ARM 进行电子设计的方法。在本书的后半部分介绍了基于 ARM 的网络传感器及其应用以及 SAMSUNG 的 ARM 控制器 S3C4510B。书中的电路图和源程序已经过实例验证,读者可以直接将其应用于自己的设计中。本书的特点是强调实用性和先进性,力求通俗易懂。

本书适用于计算机、电子、控制及信息等相关专业的在校大学生,对于广大工程技术人员,本书也具有一定的实用价值。

### 图书在版编目(CIP)数据

ARM 嵌入式系统设计及实践 / 杨恒主编. —西安: 西安电子科技大学出版社, 2005.10

ISBN 7-5606-1582-1

I. A… II. 杨… III. 微处理器, ARM—系统设计 IV. TP332

中国版本图书馆 CIP 数据核字(2005)第 112765 号

策 划 臧延新

责任编辑 王晓杰 臧延新

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

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 16.25

字 数 379 千字

印 数 1~4000 册

定 价 25.00 元

ISBN 7-5606-1582-1/TP·0908

**XDUP 1873001-1**

\*\*\*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。

## 《ARM 嵌入式系统设计及实践》编委会

- 主 编** 国际数码及嵌入式技术认证机构([www.jeeetc.com](http://www.jeeetc.com)) 杨恒(博士)
- 副主编** 武汉大学国际软件学院院长 周怀北(教授, 博士生导师)  
西北工业大学计算机学院院长 周兴社(教授, 博士生导师)  
西安交通大学软件学院常务副院长 曾 明(教授)  
西安电子科技大学软件学院院长 武 波(教授)  
北京大学深圳研究生院信息学院副院长 王新安(副教授)  
广州南沙资讯科技园 唐会华(工程师)
- 编 委** 王旭辉 许海港 杨秀芳 胡 燕 田国华

# 前 言

当今世界,科技飞速发展,人类面临着一次又一次的机遇和挑战。日常生活中,各种使用嵌入式技术的电子产品已经得到了广泛的应用,如MP3、PDA、手机、智能玩具、智能家电、车载电子设备等。在工业和服务领域中,嵌入式技术也已经大量应用于数控机床、智能工具、机器人等行业,正在逐渐改变着传统的工业生产和服务方式。据统计,每年只有10%~20%的计算机芯片是为台式PC机或手提电脑而设计的;为嵌入式系统设计制造的CPU每年大概有10~20亿片;每年又有10 000个新的嵌入式系统计划产生,而且越来越多的控制系统需要复杂的嵌入式操作系统。

本书的作者都是长期从事嵌入式技术(ARM)研究与普及的科技人员。本书系统、全面地介绍了ARM及嵌入式系统的原理、开发环境、编程语言和应用实例,并详细地反映了该领域的最新发展成果。本书的特点是通俗易懂,深入浅出,可作为高等院校相关专业学生的教材,也可作为中高级研究人员的有益助手和参考资料。

本书已被国际数码及嵌入式技术认证机构([www.IEETC.com](http://www.IEETC.com))指定为嵌入式系统(ARM)及操作系统证书(编号IDETC071501)推荐教材。读者可以根据自身的情况与本书作者专门研发的不同版本的配套学习实验开发箱(含开发学习板、下载线缆、手册和光盘)结合使用(详见[www.edtyang.com](http://www.edtyang.com))。

作者谨向以下关心过本书或在学术上给予过帮助的朋友表示感谢:

新加坡南洋理工大学(Nanyang Technological University, Singapore) Prof.Ser

Wee, Prof.C.ZHU, Prof.G.B.HUANG;

美国密西根大学 Sven E Widmalm 教授;

亚洲咨询公司、美国密西根大学 Chris Koh 博士;

新加坡国家咨讯技术研究院林晓博士、姜力军博士、陈建峰博士;

北京大学深圳研究院赵勇先生；  
北京大学柳翔教授；  
山东大学张承瑞教授、林明星教授；  
武汉大学国际软件学院秦中先生；  
西安市科技局副局长陈长春博士；  
西安软件园毛爱亮主任、陈峻峰副主任、朱立明副主任；  
国家集成电路西安产业化基地蔺建文主任、何晓宁部长和王坤元经理；  
西安国家 863 软件基地楼文晓主任；  
美国德州仪器上海办事处张浩先生；  
西北工业大学陈明教授；  
西安理工大学李言教授、邱宗明教授、于殿泓博士；  
深圳熙和技术公司毛周明总裁、申凌博士和陶明博士；  
广州华意电子科技有限公司刘林先生；  
国际数码及嵌入式技术机构的李伟、陈翠华、张宇、毛伟云；  
武汉大学计算机学院杨汉坤；  
西安西雅图数码科技有限公司胡松辉、屈庆友；  
西安邮电学院朱自祥教授；  
北京大学崔小乐博士后；  
兰州高等专科学校李贵山教授。

由于作者水平有限，错误和不当之处在所难免，敬请各位不吝赐教。  
作者 E-mail: etdyang@163.com.

杨 恒  
2005.8

# 目 录

<b>第 1 章 ARM 系统结构</b> .....	1
1.1 ARM 简介.....	1
1.1.1 ARM 公司简介.....	1
1.1.2 产品介绍.....	1
1.2 ARM 的版本.....	2
1.3 ARM 编程模型.....	3
1.3.1 ARM 微处理器介绍.....	4
1.3.2 ARM 微处理器的工作状态.....	7
1.3.3 指令长度及数据类型.....	7
1.3.4 ARM 体系结构的存储器格式.....	8
1.3.5 处理器工作模式.....	8
1.3.6 寄存器组织.....	9
1.3.7 异常处理.....	14
1.4 本章小结.....	18
<b>第 2 章 ARM 指令系统</b> .....	19
2.1 ARM 指令系统概要.....	19
2.2 ARM 指令集分类说明.....	21
2.2.1 跳转指令.....	21
2.2.2 数据处理指令.....	22
2.2.3 乘法指令与乘加指令.....	26
2.2.4 程序状态寄存器访问指令.....	28
2.2.5 加载/存储指令.....	29
2.2.6 批量数据加载/存储指令.....	31
2.2.7 数据交换指令.....	31
2.2.8 移位指令(操作).....	32
2.2.9 协处理器指令.....	33
2.2.10 异常产生指令.....	35
2.3 ARM 指令集列表.....	36
2.4 Thumb 指令集简介.....	38
2.4.1 Thumb 指令集概要.....	38
2.4.2 Thumb 指令集列表.....	40
2.5 ARM 指令寻址方式.....	41

2.6	ARM 伪操作 .....	43
2.7	ARM 汇编语言伪指令 .....	52
2.8	本章小结 .....	54
<b>第 3 章</b>	<b>嵌入式实时操作系统 <math>\mu</math>C/OS-II .....</b>	<b>55</b>
3.1	$\mu$ C/OS-II 简介 .....	55
3.2	$\mu$ C/OS-II 内核结构 .....	56
3.2.1	临界段 .....	56
3.2.2	任务 .....	56
3.2.3	任务状态 .....	57
3.2.4	任务控制块(Task Control Blocks, OS_TCBs) .....	58
3.2.5	就绪表(Ready List) .....	59
3.2.6	任务调度(Task Scheduling) .....	60
3.2.7	给调度器上锁和开锁 .....	61
3.2.8	统计任务 .....	62
3.2.9	$\mu$ C/OS-II 中的中断处理 .....	64
3.3	$\mu$ Clinux 和 $\mu$ Clinux 引导程序 .....	66
3.3.1	Linux 的启动过程 .....	67
3.3.2	$\mu$ Clinux 的启动过程 .....	68
3.4	$\mu$ Clinux 的进程与线程分析 .....	71
3.4.1	Linux 系统对进程、线程的管理 .....	71
3.4.2	$\mu$ Clinux 的多进程处理 .....	72
3.5	$\mu$ Clinux 的文件系统分析及其构造 .....	72
3.6	$\mu$ Clinux 的开发环境 .....	75
3.7	本章小结 .....	75
<b>第 4 章</b>	<b>用 LPC2210 开发 ARM 程序实例 .....</b>	<b>77</b>
4.1	LPC2210 芯片的介绍 .....	77
4.2	开发平台 .....	89
4.2.1	SeaARM 5.0 开发板 .....	89
4.2.2	JTAG 仿真器性能介绍和特性 .....	90
4.2.3	ARM7TDMI-S 处理器 .....	92
4.3	用延时控制蜂鸣器的设计 .....	93
4.3.1	LPC2200 专用工程模板 .....	93
4.3.2	使用 LPC2200 专用工程模板建立工程 .....	94
4.3.3	编写应用程序 .....	96
4.3.4	调试运行 .....	101
4.3.5	脱机运行 .....	102
4.4	用 GPIO 模拟 SPI 总线驱动八个 LED 流水灯 .....	104



4.5 应用 LPC2210 控制图形点阵型液晶显示器.....	107
4.5.1 图形点阵型 LCD 原理介绍.....	107
4.5.2 应用 LPC2210 实现对 HS12864-10A 的显示控制.....	112
4.6 本章小结.....	121
<b>第 5 章 基于实时操作系统的应用实例.....</b>	<b>123</b>
5.1 $\mu$ C/OS-II 实时内核.....	123
5.1.1 临界段(Critical Sections).....	124
5.1.2 任务.....	124
5.1.3 任务状态.....	125
5.1.4 空闲任务(Idle Task).....	127
5.1.5 统计任务.....	127
5.1.6 $\mu$ C/OS-II 中的中断处理.....	128
5.1.7 时钟节拍.....	128
5.1.8 $\mu$ C/OS-II 的初始化.....	129
5.1.9 $\mu$ C/OS-II 的启动.....	131
5.1.10 获取当前 $\mu$ C/OS-II 的版本号.....	133
5.2 LPC2210 实时时钟(RTC).....	134
5.3 基于 $\mu$ C/OS-II 的实例分析.....	136
5.4 多任务工程.....	141
5.4.1 工程创建及片外调试.....	141
5.4.2 固化程序.....	143
5.5 LPC2210 的 I <sup>2</sup> C 总线接口.....	144
5.5.1 特性.....	144
5.5.2 I <sup>2</sup> C 总线工作原理.....	145
5.5.3 I <sup>2</sup> C 串行 EEPROM 的相关知识.....	149
5.5.4 SeaARM3.0 系统中 LPC2210 的 I <sup>2</sup> C 接口电路.....	150
5.5.5 基于 $\mu$ C/OS-II 的源程序分析.....	151
5.5.6 基于 $\mu$ C/OS-II 的 I <sup>2</sup> C 总线驱动中间件实验.....	161
5.5.7 固化程序.....	164
5.6 本章小结.....	165
<b>第 6 章 基于 ARM 的网络传感器.....</b>	<b>167</b>
6.1 网络传感器系统简介.....	167
6.2 基于 ARM 的网络传感器硬件设计.....	167
6.2.1 系统方案设计.....	167
6.2.2 网络接口设计.....	171
6.3 ARM 网络传感器的软件设计.....	174
6.3.1 嵌入式实时操作系统.....	175

6.3.2 嵌入式 TCP/IP 协议栈 .....	179
6.4 本章小结 .....	186
<b>第 7 章 SAMSUNG 4510B 控制器应用实例</b> .....	<b>187</b>
7.1 S3C4510B 功能简介 .....	187
7.1.1 S3C4510B 结构描述 .....	187
7.1.2 S3C4510B 的运行模式及转换实例 .....	189
7.2 S3C4510B 特殊功能寄存器介绍 .....	194
7.3 S3C4510B 外部总线转换方式 .....	199
7.4 系统内存映射 .....	199
7.4.1 系统管理特殊寄存器介绍 .....	199
7.4.2 系统内存映射配置 .....	202
7.4.3 外部 I/O 块时序分析 .....	204
7.4.4 可缓冲和非缓冲地址介绍 .....	207
7.5 以太网控制器介绍 .....	208
7.6 定时器 .....	209
7.6.1 定时器操作详解 .....	209
7.6.2 定时器特殊寄存器描述 .....	210
7.7 可编程 I/O 口 .....	211
7.7.1 可编程 I/O 特殊寄存器 .....	211
7.7.2 可编程 I/O 实例 .....	213
7.8 中断控制器 .....	216
7.8.1 中断源 .....	216
7.8.2 中断控制器特殊寄存器 .....	217
7.8.3 实例描述 .....	217
7.9 UART 控制器 .....	224
7.9.1 UART 框图 .....	225
7.9.2 特殊寄存器 .....	225
7.9.3 应用实例 .....	227
7.10 本章小结 .....	230
<b>附录一 ADS 集成开发环境的使用</b> .....	<b>231</b>
<b>附录二 SeaARM 5.0 学习开发工具</b> .....	<b>245</b>
<b>参考文献</b> .....	<b>249</b>

# 第 1 章 ARM 系统结构

## 1.1 ARM 简介

### 1.1.1 ARM 公司简介

ARM(Advanced RISC Machines)公司是微处理器行业的一家知名企业,其主要产品有大量高性能、低成本、低能耗的 RISC 处理器及其相关的技术和软件。RISC 处理器适用于多种领域,比如嵌入式控制系统、消费/教育类多媒体、DSP 和移动式应用等。

ARM 将其技术授权给世界上许多著名的半导体、软件和 OEM 厂商,每个厂商得到的都是一套独一无二的 ARM 的相关技术及服务。利用这种合作关系,ARM 很快成为了许多全球性 RISC 标准的缔造者。

目前,总共有 30 多家半导体公司与 ARM 签订了硬件技术使用许可协议,其中包括 Intel、IBM、LG、NEC、SONY、Philips 这样的大公司。至于其软件系统的合伙人,则包括微软、升阳和 MRI 等一系列知名公司。

ARM 架构是面向低预算市场设计的第一款 RISC 微处理器。

### 1.1.2 产品介绍

ARM 提供了一系列内核、扩展体系、微处理器和系统芯片方案。由于所有产品均采用一个通用的软件体系,因而相同的软件可在所有产品中运行(理论上如此)。典型的产品如下:

#### 1) CPU 内核

① ARM7: 小型、快速、低能耗、集成式 RISC 内核,用于移动通信。

② ARM7TDMI(Thumb): 这是公司授权用户最多的一项产品,将 ARM7 指令集同 Thumb 扩展组合在一起,以减少内存容量和系统成本。同时,它还利用嵌入式 ICE 调试技术来简化系统设计,并用一个 DSP 增强扩展来改进性能。该产品的典型用途是数字蜂窝式电话和硬盘驱动器。

③ ARM9TDMI: 采用五阶段管道化 ARM9 内核,同时配备 Thumb 扩展、调试和 Harvard 总线。在生产工艺相同的情况下,性能参数可达到 ARM7TDMI 的两倍之多。

#### 2) 体系扩展

Thumb 以 16 位系统的成本,提供 32 位 RISC 的性能,特别值得一提的是它所需的内存容量非常小。

### 3) 嵌入式 ICE 调试

由于集成了类似于 ICE 的 CPU 内核调试技术,因而原型设计和系统芯片的调试得到了极大的简化。

### 4) 微处理器

① ARM710 系列:包括 ARM710、ARM710T、ARM720T 和 ARM740T,它们具有低成本、低能耗、封装式的常规系统微型处理器,并配有高速缓存(Cache)、内存管理、写缓冲和 JTAG。该系列广泛地应用于手持式计算设备、数据通信设备和消费类多媒体中。

② ARM940T、920T 系列:该系列具有低成本、低能耗、高性能的系统微处理器,并配有 Cache、内存管理和写缓冲。该系列应用于高级引擎管理、保安系统、顶置盒、便携式计算机和高档打印机。

③ StrongARM:它是一种性能很高、同时满足常规应用需要的微处理器技术,与 DEC 联合研制,后来授权给 Intel。SA110 处理器、SA1100 PDA 系统芯片和 SA1500 多媒体处理器芯片均采用了这一技术。

④ ARM7500 和 ARM7500FE:它们是高度集成的单芯片 RISC,基于一个缓存式 ARM7 32 位内核,拥有内存和 I/O 控制器、三个 DMA 通道、片上视频控制器和调色板以及立体声端口。ARM7500FE 则增加了一个浮点运算单元以及对 EDO DRAM 的支持。该系列特别适合于电视顶置盒和网络计算机(NC)。

Windows CE 的 Pocket PC 只支持 ARM,而 Windows CE 可支持多种嵌入式处理器,但基于 Windows CE 的 Pocket PC 则只支持 ARM 一种。微软在对 SH3、MIPS、ARM 等嵌入式处理器做了评估后认为,ARM 系列是一种性价比较好的产品。目前 ARM 在手持式设备市场占有 90%以上的份额,只支持 ARM,可以有效地缩短应用程序开发与测试的时间,也降低了研发费用。由于 ARM 开放了其处理器授权,因此,用户可以在市场上多家整机厂商中进行选择,从而保证了这一市场的竞争性。

## 1.2 ARM 的版本

到目前为止,ARM 体系结构共定义了六个版本,版本号分别为 1~6。从版本 1 到版本 6,ARM 体系的指令功能不断扩大。同时,ARM 处理器系列中的各种处理器所采用的实现技术各不相同,性能差别很大,应用场合也有所不同,但是只要它们支持相同的 ARM 体系版本,给予它们的应用软件是兼容的。ARM 体系结构的六个版本的特点分别如下。

### 1) 版本 1

因为其内核的功能比较简单,所以没有在商业产品中使用。本版本中地址空间是 26 位,目前已经不再使用。它包含下列指令:

- ① 除乘法指令之外的基本数据处理指令。
- ② 集于字节、字和多字的读取和写入指令。
- ③ 包括子程序调用指令 BL 在内的跳转指令。
- ④ 供操作系统使用的软件中断指令。

### 2) 版本 2

与版本 1 相比,版本 2 增加了下列指令:

- ① 乘法指令和乘加法指令。
- ② 支持协处理器的指令。
- ③ 对于 FIQ 模式，提供了额外的两个备份寄存器。
- ④ SWP 指令及 SWPB 指令。

本版本中地址空间是 26 位，目前已经不再使用。

### 3) 版本 3

版本 3 较以前的版本发生了比较大的变化，主要的改进部分如下：

- ① 处理器的地址空间扩展到了 32 位，同时也支持 26 位的地址空间。
- ② 当前程序状态信息从原来的 R15 寄存器移到一个新的寄存器中，新寄存器名为当前程序状态寄存器 CPSR(Current Program Status Register)。
- ③ 增加了备份的程序状态寄存器 SPSR(Saved Program Status Register)，用于在程序异常中断时，存储被中断程序的程序状态。
- ④ 增加了两个处理模式，使操作系统代码可以方便地使用数据访问中止异常、指令预取中止异常和未定指令异常。
- ⑤ 增加了 MRS 和 MSR 指令，用于访问 CPSR 寄存器和 SPSR 寄存器。
- ⑥ 修改了原来的从异常中返回的指令。

### 4) 版本 4

与版本 3 比较，版本 4 增加了下列指令：

- ① 半字的读取和写入指令。
- ② 读取带符号的字节和半字节数据指令。
- ③ 增加了 T 变种，可以使处理器状态切换到 Thumb 状态，在该状态下指令集是 16 位的 Thumb 指令集。
- ④ 增加了处理器的特权模式。在该模式下，使用的是用户模式下的寄存器。

### 5) 版本 5

- ① 提高了 T 变种中的 ARM/Thumb 混合使用的效率。
- ② 对于 T 变种的指令和非 T 变种的指令使用相同的代码生成技术。
- ③ 增加了前导零计数指令，该指令可以使整数除法和中断优先级排队操作都有效。
- ④ 增加了软件断点指令。
- ⑤ 为协处理器设计提供了更多的可选择的指令。
- ⑥ 增加了严格定义的乘法指令对条件标志位的影响。

### 6) 版本 6

ARM 体系版本 6 是 2001 年发布的，其主要特点是增加了 SIMD 功能扩展。它适合于使用电池供电的高性能的便携式设备。

## 1.3 ARM 编程模型

本节介绍 ARM 微处理器编程模型的一些基本概念，包括 ARM 微处理器、工作状态切换、数据的存储格式、处理器异常等。通过对本节的阅读，希望读者能够了解 ARM 微处

理器的基本工作原理和一些与程序设计相关的基本技术细节,为以后的程序设计打下基础。

### 1.3.1 ARM 微处理器介绍

ARM 微处理器目前包括六个系列产品 ARM7、ARM9、ARM9E、ARM10E、SecurCore 和最新的 ARM11 系列。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以外,每一个系列的 ARM 微处理器都有各自的特点和应用领域。

下面我们介绍 ARM7、ARM9、ARM11 系列。

#### 1. ARM7 系列

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

- ① 具有嵌入式 ICE-RT 逻辑,调试开发方便。
- ② 极低的功耗,适合于对功耗要求较高的应用,如便携式设备。
- ③ 能够提供 0.9 MIPS/MHz 的三级流水线结构。
- ④ 代码密度高并兼容 16 位的 Thumb 指令集。
- ⑤ 对操作系统的支持广泛,包括 Windows CE、Linux、Palm OS 等。
- ⑥ 指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容,便于用户的产品升级换代。
- ⑦ 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 系列微处理器的主要应用领域为:工业控制、Internet 设备、网络设备、移动电话等多种多媒体和嵌入式应用。

ARM7 系列微处理器包括四种类型的核: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 的基本含义为:

- 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. ARM7 微处理器的寄存器结构

ARM7 微处理器共有 37 个寄存器,它们被分为若干个组(Bank)。这些寄存器包括:

- ① 31 个通用寄存器,包括程序计数器(PC 指针),均为 32 位的寄存器。
- ② 六个状态寄存器,用以标识 CPU 的工作状态及程序的运行状态,均为 32 位,目前只使用了其中的一部分。

同时,ARM 处理器又有七种不同的处理器模式,在每一种处理器模式下均有一组相应的寄存器与之对应。即在任意一种处理器模式下,可访问的寄存器包括 15 个通用寄存器(R0~R14)、一个或两个状态寄存器和程序计数器。在所有的寄存器中,有些是在七种处理器模式下共用同一个物理寄存器,而有些寄存器则是在不同的处理器模式下有不同的物理寄存器。

关于 ARM 处理器的寄存器结构,在后面的相关章节中将会详细介绍。

### 3. ARM7 微处理器的应用选型

随着国内外嵌入式应用领域的逐步发展, ARM 微处理器必然会得到广泛的重视和应用。但是, 由于 ARM 微处理器有多达几十种的内核结构, 几十个芯片生产厂家, 以及千变万化的内部功能配置组合, 给开发人员在选择方案时带来了一定的困难, 所以对 ARM 芯片做一些对比研究是十分必要的。

下面从应用的角度出发, 对在选择 ARM7 微处理器时所应考虑的主要问题做一些简要的探讨。

#### 1) ARM7 微处理器内核的选择

从前面所介绍的内容可知, ARM 微处理器包含一系列的内核结构, 以适应不同的应用领域。用户如果希望使用 Windows CE 或标准 Linux 等操作系统以减少软件开发时间, 就需要选择 ARM720T 以上带有 MMU(Memory Management Unit)功能的 ARM 芯片, 其中 ARM720T、ARM920T、ARM922T、ARM946T、Strong-ARM 都带有 MMU 功能。而 ARM7TDMI 则没有 MMU, 不支持 Windows CE 和标准 Linux, 但目前有  $\mu$ CLinux 等不需要 MMU 支持的操作系统可运行于 ARM7TDMI 硬件平台之上。事实上,  $\mu$ CLinux 已经能够成功移植到多种不带 MMU 的微处理器平台上, 并在稳定性和其他方面都有上佳的表现。

#### 2) 系统的工作频率

系统的工作频率在很大程度上决定了 ARM 微处理器的处理能力。ARM7 系列微处理器的典型处理速度为 0.9 MIPS/MHz, 常见的 ARM7 芯片系统主时钟为 20~133 MHz; ARM9 系列微处理器的典型处理速度为 1.1 MIPS/MHz, 常见的 ARM9 的系统主时钟频率为 100~233 MHz; ARM10 最高可以达到 700 MHz。不同芯片对时钟的处理不同, 有的芯片只需要一个主时钟频率, 有的芯片内部时钟控制器可以分别为 ARM 核、USB、UART、DSP、音频等功能部件提供不同频率的时钟。

#### 3) 芯片内存储器的容量

大多数的 ARM 微处理器片内存储器的容量都不太大, 需要用户在设计系统时外扩存储器, 但也有部分芯片具有相对较大的片内存储空间, 如 ATMEL 的 AT91F40162 就具有高达 2 MB 的片内程序存储空间。用户在设计时可考虑选用这种类型, 以简化系统的设计。

#### 4) 片内外围电路的选择

除 ARM 微处理器核以外, 几乎所有的 ARM 芯片均根据各自不同的应用领域, 扩展了相关的功能模块, 并集成在芯片之中, 我们称之为片内外围电路, 如 USB 接口、LCD 控制器、键盘接口、RTC、ADC 和 DAC、DSP 协处理器等。设计者应分析系统的需求, 尽可能采用片内外围电路完成所需的功能。这样既简化了系统的设计, 同时也提高了系统的可靠性。

### 4. ARM9 系列

ARM9 系列处理器使用的是 ARM9TDMI 处理器核, 其中包含了 16 位的 Thumb 指令集。使用 Thumb 指令集可以以 16 位的系统开销得到 32 位的系统性能。

ARM9 系列包含 ARM920T、ARM922T、ARM940T 三种类型, 主要用于适应不同的市场需求。

ARM9 系列处理器具体应用于以下场合:

- ① 下一代无线设备，包括视频电话和 PDA 等。
- ② 数字消费品，包括机顶盒、网关等。
- ③ 成像设备，包括打印机、数字照相机和数字摄像机等。
- ④ 汽车、通信和信息系统。

ARM9 系列处理器具有以下主要特点：

- ① 支持 32 位 ARM 指令集和 16 位 Thumb 指令集的 32 位 RISC 处理器。
- ② 五级整数流水线。
- ③ 单一的 32 位 AMBA 总线接口。
- ④ MMU 支持 Windows CE、PalmOS、SymbianOS、Linux 等。
- ⑤ MPU 支持实时嵌入式操作系统。
- ⑥ 统一的数据 Cache 和指令 Cache。
- ⑦ 提供 0.18  $\mu\text{m}$ 、0.15  $\mu\text{m}$  及 0.13  $\mu\text{m}$  的生产工艺。

ARM9 产品性能见表 1-1。

表 1-1 ARM9 产品性能表

ARM9 性能特性							
型号	Cache 大小 (指令/数据)	紧密耦合	存储器管理存储器 (TCM)	AiIB	Thumb 总线	DSP	Jazllc
ARM920T	16 K/16 K	无	MMU	有	有	无	无
ARM922T	8 K/8 K	无	MMU	有	有	无	无
ARM940T	4 K/4 K	无	MMU	有	有	无	无

### 5. ARM11 系列

ARM 公司最近公布了四个新的 ARM11 系列微处理器内核(ARM1156T2-S 内核、ARM1156T2F-S 内核、ARM1176JZ-S 内核和 ARM11JZF-S 内核),以及应用于 ARM1176JZ-S、ARM11JZF-S 内核系列的 PrimeXsys 平台和相关的 CoreSight 技术。

ARM1156T2-S 和 ARM1156T2F-S 内核都基于 ARMv6 指令集体系结构,将是首批含有 ARM Thumb-2 内核技术的产品,可令合作伙伴进一步减少与存储系统相关的生产成本。两款新内核主要用于多种嵌入式存储器、汽车网络和成像应用产品。它们提供了更高的 CPU 性能和吞吐量,并增加了许多特殊功能,可解决新一代装置的设计难题。这两款新内核在体系结构中增加的功能包括:对于汽车安全系统类安全应用产品的开发至关重要的存储器容错能力。ARM1156T2-S 和 ARM1156T2F-S 内核与新的 AMBA 3.0 AXI 总线标准一致,可满足高性能系统的大量数据存取需求。Thumb-2 内核技术结合了 16 位、32 位指令集体系结构,提供了更低的功耗、更高的性能、更短的编码。该技术提供的软件技术方案使用的存储空间较现用的 ARM 技术方案减少了 26%;较现用的 Thumb 技术方案速度增快了 25%。

ARM1176JZ-S 和 ARM1176JZF-S 内核及 PrimeXsys 平台是首批以 ARM TrustZone 技术实现手持式装置和消费类电子装置中公开操作系统的超强安全性的产品,同时也是首次对可节约高达 75% 处理器功耗的 ARM 智能能量管理(ARM Intelligent Energy Manager)进行一体化支持。ARM1176JZ-S 和 ARM1176JZF-S 内核基于 ARMv6 指令集体系结构,主要为供应商和运营商所提供的新一代消费类电子装置的电子商务和安全的网络下载提供支持服务。



ARM1176JZ-S 和 ARM1176JZF-S 内核、PrimeXsys 平台提供了安全的低功耗设计。它们所含的 AMBA 3.0 AXI 可对频率和电压变化进行控制,同时它们还可进行系统级 TrustZone 软硬件参考设计。两个新内核中集成了 ARM Jazelle 技术,可加快嵌入式 Java 的执行。ARM1176JZF-S 内核包含一个浮点协处理器,非常适合用于嵌入式 3D 图像应用产品。两个新内核的标准配置中都含有 ARM-Synopsys RTL to GDSII 参考技术方案,都是可综合的。在 0.13  $\mu\text{m}$  工艺中,最低频率可达 333~550 MHz。PrimeXsys 平台包含 ARM CoreSight 技术,提供了世界领先的调试和跟踪技术方案。

ARM CoreSight 技术可快速地对不同的软件进行调试,并可对多核和 AMBA 总线的情况进行同时跟踪。此外,还可同时对多核进行调试,CoreSight 技术可对 AMBA 上的存储器和外设进行调试,无需暂停处理器工作,达到了一般不易做到的实时开发。ARM CoreSight 技术拥有更高的压缩率,为半导体制造商们提供了对新的更高频处理器进行调试、跟踪的技术方案。使用 CoreSight 技术,制造商们可通过减少调试所需的管脚、减少片上跟踪缓存所需的芯片面积等手段来降低生产成本。

### 1.3.2 ARM 微处理器的工作状态

从编程的角度来看,ARM 微处理器的工作状态一般有两种,并可在两种状态之间进行切换。这两种状态分别是:

- ① ARM 状态,此时处理器执行 32 位的、字对齐的 ARM 指令。
- ② Thumb 状态,此时处理器执行 16 位的、半字对齐的 Thumb 指令。

当 ARM 微处理器执行 32 位的 ARM 指令集时,工作在 ARM 状态;当 ARM 微处理器执行 16 位的 Thumb 指令集时,工作在 Thumb 状态。在程序的执行过程中,微处理器可以随时在两种工作状态之间切换,并且处理器工作状态的转变并不影响处理器的工作模式和相应寄存器中的内容。

ARM 指令集和 Thumb 指令集均有切换处理器状态的指令,并可在两种工作状态之间进行切换,但 ARM 微处理器在开始执行代码时,应该处于 ARM 状态。两种状态之间的切换方法如下:

① 进入 Thumb 状态:当操作数寄存器的状态位(位 0)为 1 时,可以采用执行 BX 指令的方法,使微处理器从 ARM 状态切换到 Thumb 状态。此外,当处理器处于 Thumb 状态时发生异常(如 IRQ、FIQ、Undef、Abort、SWI 等),则异常处理返回时,自动切换到 Thumb 状态。

② 进入 ARM 状态:当操作数寄存器的状态位为 0 时,执行 BX 指令时可以使微处理器从 Thumb 状态切换到 ARM 状态。此外,在处理器进行异常处理时,把 PC 指针放入异常模式链接寄存器中,并从异常向量地址开始执行程序,也可以使处理器切换到 ARM 状态。

### 1.3.3 指令长度及数据类型

ARM 微处理器的指令长度可以是 32 位(在 ARM 状态下),也可以为 16 位(在 Thumb 状态下)。ARM 微处理器中支持字节(8 位)、半字(16 位)、字(32 位)三种数据类型。其中,字需要 4 字节对齐(地址的低两位为 0)、半字需要 2 字节对齐(地址的最低位为 0)。有些 ARM