



普通高校“十二五”规划教材

ARM

ARM 处理器

C

语言开发应用 (第2版)

范书瑞 赵燕飞 高铁成 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

014014517

TP332
183-2



普通高校“十二五”规划教材

ARM 处理器与 C 语言 开发应用 (第 2 版)

范书瑞 赵燕飞 高铁成 编著



TP332

183-2

北京航空航天大学出版社



北航

C1701344

内 容 简 介

在设计 ARM 控制系统时,使用 C 语言编程开发 ARM 应用程序,可以达到事半功倍的效果。本书在设计 Samsung 公司的 S3C2410A 控制系统时,启动代码采用汇编语言,应用程序大量使用 C 语言程序,大大加快了开发进度。

本书以 S3C2410A 各个模块的应用为主线,全面介绍嵌入式系统开发过程。具体内容包括:ARM 微处理器概述、ARM 编程模型和指令集介绍、ARM 程序设计基础、S3C2410A 处理器硬件结构、存储系统设计、时钟和定时器、通用端口和中断设计、串行接口设计、LCD 接口设计、ADC 和触摸屏接口设计、引导程序设计、嵌入式操作系统及其应用开发。书中从部件编程到嵌入式应用程序设计,都通过完整的实例讲解。书中所提供的程序均已在目标板上调试通过。相对于第一版,本书在版本、内容等方面进行了更新。

本书内容实用易懂,可作为嵌入式开发人员的参考书,也可作为相关专业本科生和研究生教材及参考书。

图书在版编目(CIP)数据

ARM 处理器与 C 语言开发应用 / 范书瑞等编著. --2 版

. --北京:北京航空航天大学出版社,2014. 1

ISBN 978-7-5124-1237-8

I. ①A… II. ①范… III. ①微处理器—系统设计②
C 语言—程序设计 IV. ①TP332 ②TP312

中国版本图书馆 CIP 数据核字(2013)第 198873 号

版权所有,侵权必究。

ARM 处理器与 C 语言开发应用(第 2 版)

范书瑞 赵燕飞 高铁成 编著

责任编辑 董立娟

*

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(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:22.25 字数:474 千字

2014 年 1 月第 2 版 2014 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1237-8 定价:49.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

第 2 版前言

迄今为止,ARM 公司向 250 多家公司出售了 800 个处理器许可证,已生产超过 250 亿个处理器,每天销量超过 1 600 万。ARM 技术在 90% 的智能手机、80% 的数码相机以及 28% 的电子设备中得到应用,已经成为嵌入式应用和设计的主角,是真正意义上的数字世界的架构(The Architecture for the Digital World)。目前 ARM 处理器主流产品分为 Cortex - A、Cortex - R 和 Cortex - M 三大系列,经典的 ARM9 处理器依旧在教学中占有重要地位,同时随着 ARM 公司所推出的 Microcontroller Development Kit 开发环境,使得经典 ARM 应用开发更加便捷。

本书在第一版的基础上对部分章节做了修改:

第 1 章 结合最新 ARM 技术的发展,增加了 ARMv8 体系结构和 Cortex 系列的介绍,同时更新了 ARM 处理器选型。

第 3 章 修改了“3.5 程序设计示例”,添加“3.6 MDK 和配置向导”和“3.7 微处理器软件架构工具”。

第 7 章 修改了“7.1.4 端口程序设计”和“7.2.5 外部中断程序设计”。

第 11 章 将 11.3 节更新为引导程序配置向导分析。

另,部分程序也进行了修改。

编 者
2013 年 12 月

前言

ARM 公司自 1990 年 11 月正式成立以来,在 32 位 RISC 开发领域不断取得突破。截至 2006 年 9 月,ARM 公司与合作伙伴所签订的处理器授权协议总数达到 448 项,物理 IP 的授权协议总数达到 263 项。在所有 IP 授权市场中,ARM 占有 26%,份额远超过其他竞争对手。随着 ARM 处理器在全球范围的流行,32 位的 RISC 嵌入式处理器已经成为嵌入式应用和设计的主流。

目前,随着嵌入式技术的不断发展,国内外越来越多的工程师都选用 ARM 来设计他们的产品,ARM9 处理器在嵌入式系统开发中占有重要位置。本书以 S3C2410A 微处理器为硬件平台,在简要介绍汇编语言的基础上,详细讲解基于 ARM 处理器的 C 语言开发技术。

第 1 章 简要介绍 ARM 处理器的基本概念,以及 ARM 处理器扩展技术、类型和处理器的选型。

第 2 章 介绍 ARM 处理器编程模型和指令集知识,包括 ARM 编程模型、ARM 指令结构和寻址方式,以及 ARM 指令集、Thumb 指令集和 DSP 扩展指令集。通过本章的阅读,可使读者了解 ARM 汇编程序的开发。

第 3 章 介绍 ARM 程序设计基础知识,着重介绍汇编和 C 语言的混合编程技术。包括伪操作、伪指令、汇编,以及 C 语言的混合编程技术和编译器的使用,均通过实例进行了详细的讲解。通过本章的阅读,读者能够开发简单的 C 语言和汇编混合程序。

第 4 章 介绍 S3C2410A 处理器的硬件结构。通过对本章的阅读,读者可以掌握基于 S3C2410A 的最小系统设计。

第 5 章 介绍 S3C2410A 的存储器设计技术,包括存储器控制寄存器、存储器地址接口设计、Nor Flash 接口设计、Nand Flash 接口设计和 SDRAM 接口设计。

第 6 章 介绍 S3C2410A 处理器的时钟和功耗管理、PWM 定时器、实时时钟、看门狗。

第 7~10 章 介绍 S3C2410A 处理器的中断通用端口设计、串行接口设计、LCD 接口设计、A/D 和触摸屏接口设计。通过阅读,读者能够掌握 S3C2410A 主要部件的编程。

第 11 章 介绍 S3C2410A 处理器引导程序设计。

第 12 章 介绍基于 S3C2410A 处理器的 Linux 移植技术,包括引导程序的移植、内核移植,以及文件系统的建立,最终在嵌入式 Linux 下成功移植 SVM,实现说话人

算法的测试。

附带程序：

各章节涉及程序都在 EDUKIT-III 实验箱上进行了调试,源代码和工程文件可以到出版社网站 www.buaapress.com.cn 的“下载专区”免费下载,其中,包括了 3.5、3.6、7.2 和 8.1 等程序完整工程文件形式,更便于读者使用。本书在第一次印刷基础上针对 MDK 开发环境进行了修改,第 3 章完善了“程序设计示例”,并添加了 MDK 配置向导规则,第 11 章通过完整的启动程序对配置向导更深入分析,还修改了第 3 章、第 7 章、第 11 章的部分程序代码。成书过程中,还得到了袁文江、董小丰的帮助,在此表示感谢。

由于编者的水平有限,书中难免存在不妥之处,恳请读者批评指正。

有兴趣的读者可以发送邮件到 fansr@hebut.edu.cn,与作者进一步交流;也可以发送邮件到 xdhydc5@sina.com,与本书策划编辑进行交流。

编 者

2011 年 2 月



第1章 ARM 微处理器概述 1

1.1 ARM 概念 1

1.2 ARM 体系结构及扩展技术 2

1.2.1 ARM 处理器的体系结构 2

1.2.2 ARM 体系结构的扩展 4

1.3 ARM 处理器系列 5

1.3.1 ARM 处理器概述 5

1.3.2 应用处理器 6

1.3.3 嵌入式处理器 9

1.3.4 专用处理器 12

1.3.5 经典 ARM 系列 12

1.4 ARM 微处理器选型 15

第2章 ARM 编程模型和指令集介绍 19

2.1 ARM 编程模型 19

2.1.1 指令长度及数据类型 19

2.1.2 ARM 处理器的工作状态 19

2.1.3 ARM 体系结构的存储器格式 20

2.1.4 处理器模式 20

2.1.5 寄存器组织 21

2.1.6 程序状态寄存器 23

2.1.7 异常 25

2.2 ARM 处理器的指令格式 28

2.2.1 ARM 处理器的指令格式 28

2.2.2 指令的条件码 29

2.3 ARM 指令寻址方式 30

2.3.1 数据处理指令的操作数寻址方式 30

2.3.2 字及无符号字节的 Load/Store 指

令的寻址方式 32

2.3.3 杂类 Load/Store 指令的寻址方式 34

2.3.4 批量 Load/Store 指令的寻址方式 34

2.3.5 协处理器 Load/Store 指令的寻址方式 36

2.4 ARM 指令集介绍 36

2.4.1 跳转指令 36

2.4.2 B 指令和 BL 指令 37

2.4.3 数据处理指令 38

2.4.4 程序状态寄存器访问指令 42

2.4.5 加载/存储指令 43

2.4.6 批量数据加载/存储指令 45

2.4.7 数据交换指令 46

2.4.8 协处理器指令 46

2.4.9 异常产生及 CLZ 指令 48

2.5 Thumb 指令及应用 48

2.6 DSP 扩展指令 49

2.6.1 带符号乘和乘加指令 49

2.6.2 饱和整数运算 50

2.6.3 存储辅助指令 52

第3章 ARM 程序设计基础 54

3.1 ARM 汇编器所支持的伪操作 54

3.1.1 符号定义伪指令 54

3.1.2 数据定义伪指令 55

3.1.3 汇编控制伪指令 56

3.1.4 其他常用的伪指令 57

3.2 汇编语言程序设计 59

3.2.1 汇编语言程序中常用的符号	60	5.1.1 存储器映像	114
3.2.2 汇编语言程序中的表达式和运算符	61	5.1.2 接口信号时序	115
3.2.3 汇编语言的程序结构	64	5.1.3 内存控制器	116
3.3 C 语言和汇编语言混合程序设计	65	5.2 地址线接口设计	122
3.3.1 汇编语言与 C/C++ 的混合编程	65	5.2.1 8 位存储器接口设计	123
3.3.2 ATPCS 规则	69	5.2.2 16 位存储器接口设计	125
3.4 编译器的使用	72	5.3 Nor Flash 接口设计	126
3.4.1 ARM 编译器 C/C++ 库	72	5.3.1 Nor Flash 和 Nand Flash 的区别	126
3.4.2 ARM 编译器关键词	73	5.3.2 SST39VF1601 电路设计	128
3.4.3 ARM 编译器支持的数据类型	74	5.4 Nand Flash 接口设计	130
3.5 程序设计示例	75	5.4.1 性能	130
3.6 MDK 配置向导	77	5.4.2 Nand Flash 控制功能寄存器	131
3.6.1 从 ADS 到 MDK 的移植	78	5.4.3 接口设计	133
3.6.2 配置向导	79	5.5 SDRAM 接口电路设计	134
3.6.3 配置举例	80	第 6 章 时钟和定时器	138
3.7 微处理器软件架构工具	85	6.1 时钟和功耗管理	138
第 4 章 S3C2410A 处理器硬件结构	88	6.1.1 锁相环	138
4.1 S3C2410A 处理器概述	88	6.1.2 时钟和功耗控制逻辑	139
4.2 引脚介绍	90	6.1.3 时钟和功耗管理寄存器	142
4.3 特殊功能寄存器	97	6.1.4 程序设计	144
4.4 最小应用系统设计	107	6.2 PWM 定时器	145
4.4.1 电源电路设计	107	6.2.1 PWM 定时器操作	145
4.4.2 晶振电路设计	108	6.2.2 PWM 定时器控制寄存器	149
4.4.3 复位电路设计	109	6.2.3 程序设计	153
4.5 JTAG 调试接口设计	110	6.3 实时时钟	153
4.5.1 JTAG 概述	110	6.3.1 RTC 功能描述	154
4.5.2 接口电路设计	111	6.3.2 RTC 特殊功能寄存器	155
4.5.3 访问处理器寄存器	113	6.3.3 RTC 程序设计	158
第 5 章 存储系统设计	114	6.4 看门狗	160
5.1 存储器控制器	114	6.4.1 看门狗操作	160
		6.4.2 看门狗寄存器	161
		6.4.3 看门狗程序设计	162

第7章 通用端口和中断设计	165
7.1 I/O 端口控制寄存器	165
7.1.1 端口寄存器	165
7.1.2 外部中断寄存器	171
7.1.3 通用状态寄存器	175
7.1.4 端口程序设计	176
7.2 S3C2410A 中断控制器	177
7.2.1 中断控制	177
7.2.2 中断源	178
7.2.3 中断控制专用寄存器	180
7.2.4 中断子控制寄存器	183
7.2.5 外部中断程序设计	184
第8章 串行接口设计	187
8.1 UART 接口及编程实例	187
8.1.1 UART 的操作	188
8.1.2 UART 特殊功能寄存器	190
8.1.3 编程实例	196
8.2 IIC 总线接口及编程实例	201
8.2.1 IIC 总线协议	201
8.2.2 S3C2410A IIC 接口操作	203
8.2.3 IIC 总线特殊功能寄存器	206
8.2.4 编程实例	208
8.3 SPI 接口及编程实例	213
8.3.1 SPI 操作	213
8.3.2 SPI 特殊功能寄存器	215
8.3.3 编程实例	218
第9章 LCD 接口设计	220
9.1 LCD 控制器的特性	220
9.2 STN LCD 控制器基本操作	221
9.2.1 视频操作	223
9.2.2 显示驱动方式	224
9.2.3 数据存储器格式	225
9.3 TFT LCD 控制器基本操作	226
9.3.1 视频操作	227
9.3.2 256 色调色板	228
9.4 LCD 控制器的使用	230
9.4.1 控制器功能寄存器	230
9.4.2 LCD 控制寄存器的初始化	237
9.5 LCD 编程实例	237
9.5.1 液晶屏初始化	237
9.5.2 显示像素	239
9.5.3 图形函数	240
9.5.4 测试程序	243
第10章 ADC 和触摸屏接口设计	244
10.1 A/D 转换器	244
10.1.1 A/D 转换步骤	244
10.1.2 ADC 的主要类型	247
10.1.3 A/D 转换器的主要技术指标	254
10.2 触摸屏原理	254
10.2.1 触摸屏结构	255
10.2.2 触摸屏的主要类型	255
10.3 ADC 和触摸屏接口	257
10.3.1 接口方式	257
10.3.2 功能描述	258
10.3.3 ADC 和触摸屏专有寄存器	260
10.4 ADC 和触摸屏接口编程实例	263
10.4.1 ADC 编程实例	263
10.4.2 触摸屏编程实例	263
第11章 引导程序设计	266
11.1 引导机理分析	266
11.1.1 引导程序工作原理	266
11.1.2 引导程序的启动过程	267
11.2 ARM9 初始化代码分析	269
11.2.1 外部文件	269
11.2.2 定义常量	272

11.2.3 异常处理	272	12.3 嵌入式 Linux 的移植	304
11.2.4 主体程序	276	12.3.1 建立交叉编译环境	304
11.2.5 调用 C 语言程序	280	12.3.2 引导程序的移植	305
11.3 引导程序配置向导分析	281	12.3.3 Linux 内核的移植	309
11.3.1 堆栈配置	282	12.3.4 根文件系统的实现	311
11.3.2 看门狗配置	284	12.4 应用程序的开发	314
11.3.3 时钟配置	285	12.4.1 Hello 测试程序	314
11.3.4 存储器配置	288	12.4.2 说话人识别算法的实现	315
11.3.5 I/O 端口配置	290	12.5 网络编程实例	318
第 12 章 嵌入式操作系统及其应		12.5.1 网络驱动程序	318
用开发	293	12.5.2 socket 基本函数	320
12.1 嵌入式操作系统概况	293	12.5.3 TCP 编程实例	325
12.1.1 嵌入式操作系统特点	293	12.5.4 综合训练之 UDP 编程实现 ...	328
12.1.2 常用嵌入式操作系统	294	附 录	331
12.2 Linux 发展支柱和开发基础 ...	296	参考文献	346
12.2.1 Linux 发展支柱	296		
12.2.2 开发工具的使用	297		

第 1 章

ARM 微处理器概述

1.1 ARM 概念

ARM 是 Advanced RISC Machines 的缩写,既可以认为是一个公司的名字,也可以认为是对一类微处理器的通称,还可以认为是一种技术的名字。

1985 年 4 月 26 日,第一个 ARM 原型在英国剑桥的 Acorn 计算机有限公司诞生,由美国加州 San Jose VLSI 技术公司制造。20 世纪 80 年代后期,ARM 很快被开发成 Acorn 的台式机产品,形成英国计算机教育的基础。1990 年成立了 ARM Limited(Advanced RISC Machines Limited,即 ARM 公司)。20 世纪 90 年代,ARM 的 32 位嵌入式 RISC(Reduced Instruction Set Computer,精简指令集计算机)扩展到世界范围,占据了低功耗、低成本和高性能的嵌入式系统应用领域的领先地位。

目前,ARM 公司专门从事基于 RISC 技术芯片的设计开发,作为知识产权供应商,本身不直接从事芯片生产,靠转让设计许可由合作公司生产各具特色的芯片。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 相关技术及服务。利用这种合作关系,ARM 很快成为许多全球性 RISC 标准的缔造者。

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在使用 ARM 公司的授权,因此既使得 ARM 技术获得更多的第三方工具、制造、软件的支持,又使整个系统成本降低,使产品更容易进入市场被消费者所接受,也更具有竞争力。

采用 ARM 技术 IP 核的微处理器,应用领域涉及无线、网络、消费娱乐、影像、汽车电子、安全应用及存储装置。ARM 公司提供的产品,包括 16/32 位 RISC 微处理器、数据引擎、三维图形处理器、数字单元库、嵌入式存储器、外设、软件、开发工具以及模拟和高速连接产品。ARM 公司协同众多技术合作伙伴为业界提供快速、稳定、完整的系统解决方案。

随着 ARM 处理器在全球范围的流行,32 位的 RISC 嵌入式处理器已经成为嵌入式应用和设计的主流。国内外越来越多的工程师在选用 ARM 技术来设计他们的

产品。

1.2 ARM 体系结构及扩展技术

迄今为止,ARM 公司定义了 8 种主要的 ARM ISA(Instruction Set Architecture,即指令集体系结构),随着版本的升级,指令集有了巨大的改进,功能不断扩大,其版本号分别为 v1~v8。同时,各版本中还有一些变种,这些变种定义了指令集中不同的功能。对于相同的 ARM 体系结构,基于它们的应用软件是兼容的。

1.2.1 ARM 处理器的体系结构

ARM 微处理器现用的体系结构中至少支持两种指令集:32 位 ARM 指令集和 16 位 Thumb 指令集。Thumb 指令集为 ARM 指令集的功能子集,但与等价的 ARM 代码相比较,使用 Thumb 指令集可以得到密度更高的代码。ARM 指令集的体系结构与扩展如图 1.1 所示。

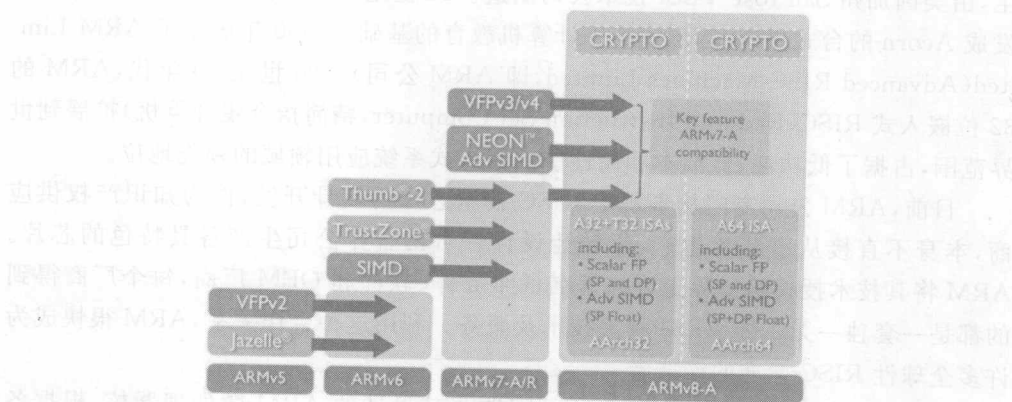


图 1.1 ARM 指令结构扩展图

1. ARMv4

ARMv4 是目前所支持的最老的版本,之前的版本不再使用。某些 ARM7 系列和 Intel 的 StrongARM 处理器采用该版本指令集。ARMv4 指令集可以在 32 位地址空间执行 32 位的 ARM 指令集。并且有了 T 变种,可以在 Thumb 状态下支持 16 位的 Thumb 指令集。这与 32 位指令相比可以节省 35% 的存储空间,且依旧保留了 32 位系统的优势。

2. ARMv5

ARMv5 是在 1999 年发布的,与 ARMv4 相比,提升了 ARM 和 Thumb 两种指令的交互工作能力,并改进了在 T 变种中 ARM/Thumb 状态之间的切换效率。同

时有了 E 变种,可以支持 DSP 指令,在音频数字信号处理中可以提高 70% 的性能。2000 年增加了 J 变种,可以运行 Java 指令。

3. ARMv6

ARMv6 版本于 2001 年发布,其主要特点是增加了 SIMD 功能扩展,同时拓展了 Thumb-2 和 TrustZone 技术。SIMD 应用于视频编解码与三维绘图等数字信号的处理中,能为音频视频在内的应用系统提供优化功能,可以使音频视频的处理能力提高 4 倍。Thumb-2 是一种新型混合指令集,融合了 16 位和 32 位指令,用于实现密度和性能的最佳平衡。作为 ARM 体系结构的扩展,TrustZone 技术是一种新的硬件安全技术。第一款 ARMv6 处理器是 ARM1136J(F)-STM,在 2002 年春季发布。

4. ARMv7

ARM v7 在相当于 ARM11 下一代的 CPU 内核 Cortex 系列上被采用,针对用途不同,定义了三大分工明确的系列。A 系列面向尖端的基于虚拟内存的操作系统和用户应用,R 系列针对实时系统,M 系列对微控制器和低成本应用提供优化。ARMv7 的 M 系列采用了 Thumb-2 技术,A 系列和 R 系列还采用了 NEON 技术,将 DSP 和媒体处理能力提高了近 4 倍,并支持改良的浮点运算,满足下一代 3D 图形和游戏应用以及传统的嵌入式控制应用的需求。

ARMv7A 注重提高运算性能,主要应用于手机、PDA、便携式游戏机等产品的微控制器。通过强化浮点运算,提高了三维图形游戏所必需的图像处理性能。ARMv7A 可以在 Windows CE、Symbian OS、Linux 等大型复杂的 OS 上运行。与 ARMv6 相比,其最大的区别就是增强了 SIMD 型指令,推出的名为 NEON 技术具有可进行 64 位数据或 128 位数据运算的混合型 SIMD 指令集。可将与 VFP 协处理器共享的寄存器按 64 位 $\times$ 32 条或 128 位 $\times$ 16 条来处理。此外,VFP 的版本已由 VFPv2 提升到了 VFPv3。

ARMv7R 重视实时处理,可以减少数据输入输出延迟时间以及提高指令预测精度等。主要面向打印机与网络终端和汽车等领域。同 ARMv7A 一样,支持 NEON 和 VFPv3。

ARMv7M 面向廉价微控制器。ARMv7A 与 ARMv7R 都是 ARMv6 的功能强化版,而 ARMv7M 则是特意降低制造成本的一种选择。既未加入 NEON 和 VFPv3 等新功能,也省略了 SIMD 指令集与缓存等,即便与 ARMv6 相比,功能也显得有些简单了。

5. ARMv8

2011 年 11 月,ARM 公司发布了其首款支持 64 位指令集的处理器架构 ARMv8。由于 ARM 处理器的授权内核广泛用于手机等诸多电子产品,ARMv8 架构作为下一代处理器的核心技术受到普遍关注。

ARMv8 在 32 位 ARM 架构上进行开发,首先用于对扩展虚拟地址和 64 位数据处理技术有更高要求的产品领域,如企业应用、高档消费电子产品。

ARMv8 架构包含两个执行状态:AArch64 和 AArch32。AArch64 执行状态针对 64 位处理技术,引入了一个全新指令集 A64;而 AArch32 执行状态将支持现有的 ARM 指令集。目前的 ARMv7 架构的主要特性,如 TrustZone 技术、虚拟化技术及 NEON advanced SIMD 技术等,都将在 ARMv8 架构中得以保留或进一步拓展。

1.2.2 ARM 体系结构的扩展

ARM 处理器体系结构在支持 32 位 ARM 指令集和 16 位 Thumb 指令集的同时,还在不同指令体系结构的基础上做了扩展。

1. Thumb 指令集

Thumb 指令集是将 ARM 指令集的一个子集重新编码而形成的一个指令集。ARM 和 Thumb 指令码分别运行于各自的处理器状态。Thumb 指令集有两个版本:版本 1 运行于 ARMv4 的 T 变种;版本 2 运行于 ARMv5 以后的 T 变种。

Thumb 指令集版本 2 通过增加指令或对已有指令的修改,提高 ARM 指令和 Thumb 指令混合使用时的效率,可以利用 Thumb 代码的密度获得 ARM 指令水平的性能。

2. 增强型 DSP 指令

增强型 DSP 指令通常称为 E 变种,首先在 ARM 体系 ARMv5TE 中使用,包含一些附加的指令,这些指令用于增强处理器对一些典型 DSP 算法的处理性能。

3. Jazelle

采用 Jazelle 技术的 J 变种首先在 ARMv5TEJ 中使用。Jazelle 是一种体系结构的扩展技术,为 ARM 处理器引入了第三套指令集——Java 虚拟机器码。Jazelle 技术提供了 Java 加速功能,可以得到比普通 Java 虚拟机高得多的性能,将 Java 的优势和先进的 32 位 RISC 芯片完美地结合在一起。

采用 Jazelle 技术与基于软件的 Java 虚拟机相比,可以提高 8 倍的性能。Jazelle 技术能在移动电话和消费电子产品等众多应用终端上显著地降低 Java 应用程序对内存的占用空间,同时提高性能并且降低功耗。

ARM 的 Jazelle 扩展集在移动设备上实现了非常优越的性能表现、快速的响应速度和顺畅的软件运行,从而给用户带来了全新的使用体验。

4. 媒体技术的扩展

SIMD(Single Instruction Multiple Data,单指令多数据)技术提供了高性能的音频/视频处理技术。ARMv7 种推出的 NEON 是一种 64/128 位单指令多数据流(SIMD)指令集,可以在低于 10 MHz 的速度下执行 MP3 音频解码功能,在 13 MHz

下运行 GSM 多速率语音编解码功能。其支持 8、16、32 和 64 位整数和单精度浮点 SIMD 运行,适合手持音频/视频处理,以及图形和游戏处理。作为 ARM 的 Optimo-DE 数据引擎的补充技术,NEON 将运用到 ARM 未来的处理器中。

5. 硬件安全技术

TrustZone 技术作为 ARM 体系结构的扩展,是一种新的硬件安全技术。ARM TrustZone 技术可识别系统的安全码和数据,硬件能清楚地区分安全信息和非安全信息。该区分能力可令安全码和数据在操作系统中安全而有效地同时运行,且不需要牺牲任何系统性能,也不会受病毒侵害。

TrustZone 技术为运行在如 Linux、Palm OS、Symbian OS、Windows CE 等开放式操作系统上的系统设备提供一种新的安全功能标准。此外,TrustZone 技术还可作为安全应用软件环境的补充,如 Sun Microsystems 的 Java 技术,令设备的运行更安全。

ARM TrustZone 技术在微处理器内核中的实现,使系统的安全功能原理设置在内核硬件中,因而去除了内核外与便携式技术无关的技术方案内容。在这种方式下,安全功能是作为一种固有特征而在所有设备的核心部分得以维护的,因此对内核及性能的影响降到了最低,同时还允许开发者在安全功能硬件基础上开发其他的安全功能。

1.3 ARM 处理器系列

1.3.1 ARM 处理器概述

ARM 是 32 位嵌入式微处理器的行业领先提供商,与业界超过 750 个合作伙伴相结合,已推出的一系列 20 多种处理器可以解决几乎所有应用难题。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(数字世界的架构)。

ARM 微处理器目前有多个系列可供选择,每一个系列的 ARM 微处理器都有各自的特点和应用领域,可以粗略划分为 ARM Cortex 应用处理器、ARM Cortex 嵌入式处理器和经典 ARM 处理器 3 大类,如图 1.2 所示。

ARM Cortex 应用程序处理器是开放式操作系统的高性能处理器 Cortex-A 系列,在高级工艺节点中可实现高达 2 GHz+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种类,最多提供 4 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。

ARM Cortex 嵌入式处理器包括 Cortex-R 和 Cortex-M 两个系列,旨在为各种不同的市场提供服务。Cortex-M 系列面向具有确定性的微控制器应用的、成本

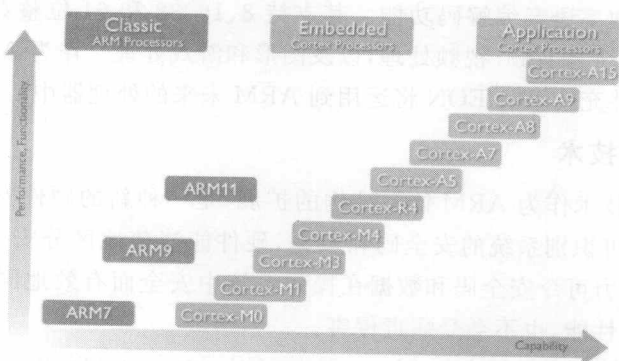


图 1.2 ARM 处理器概况图

敏感型解决方案,主要是针对微控制器领域开发的,该领域既需要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。而面向实时应用的卓越性能 Cortex-R 系列处理器的开发则面向深层嵌入式实时应用,对低功耗、良好的中断行为、卓越性能以及与现有平台的高兼容性这些需求进行了平衡考虑。Cortex-M 系列应用于微控制器、混合信号设备、智能传感器、汽车电子和气囊,而 Cortex-R 系列主要面向汽车制动系统、动力传动解决方案、大容量存储控制器、联网和打印等领域。

经典 ARM 处理器包括基于 ARMv6 架构的高性能处理器 ARM11 系列、基于 ARMv5 架构的常用处理器 ARM9 系列和面向普通应用的经典处理器 ARM7 系列。ARM 经典处理器适用于那些希望在新应用中使用经过市场验证的技术的组织。这些处理器提供了许多的特性、卓越的能效和范围广泛的操作能力,适用于成本敏感型解决方案。这些处理器每年都有数十亿的发货量,因此可确保设计者获得最广泛的体系和资源,从而最大限度地减少集成过程中出现的问题并缩短上市时间。

此外还有面向高安全性应用的 SecurCore 和面向 FPGA 的 FPGA Cores,这两种可以称为 ARM 专用处理器,旨在满足特定市场的苛刻需求。SecurCore 处理器在安全市场中用于手机 SIM 卡和识别应用,集成了多种既可为用户提供卓越性能,又能检测和避免安全攻击的技术。面向 FPGA 构造的处理器,在保持与传统 ARM 设备兼容的同时,方便用户产品快速上市。此外,这些处理器具有独立于构造的特性,因此开发人员可以根据应用选择相应的目标设备,而不会受制于特定供应商。

1.3.2 应用处理器

可执行复杂操作系统(如 Linux、Android、WindowsCE 和 Symbian)和支持复杂图形用户界面的处理器称为应用处理器,如图 1.3 所示。此类处理器集成了内存管理单元(MMU)来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。此类处理器的应用包括智能手机、智能上网本、电子书阅读器、数字电视、家用

网关、个人导航设备等产品。传统的单核处理器既包括初级的 ARM926EJ-S, 也包括常规性能超过 2 GHz 的 Cortex-A9 处理器。而多核处理器, 例如 Cortex-A9 MPCore 处理器、Cortex-A5 MPCore 处理器和 ARM11MPCore 处理器, 则最多能在单个对称或非对称系统中实现 4 个内核, 外加一个全局中断处理程序和一个监听控制单元, 因而可提供更好的性能和可扩展性。

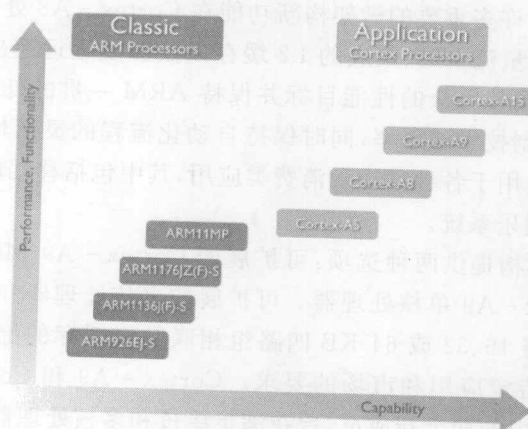


图 1.3 应用处理器

Cortex-A 处理器适用于具有高计算要求、运行丰富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。Cortex-A5、Cortex-A8、Cortex-A9 和 Cortex-A15 处理器都适用于各种不同的性能应用领域。不过, 尽管这些处理器都支持同样卓越的基础功能和完整的软件兼容性, 但提供了显著不同的特性, 可确保其完全符合未来的高级嵌入式解决方案的要求。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。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 5 000 DMIPS 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 2 000 DMIPS; Cortex-A5 低成本实现在 400~800 MHz 的频率下, 提供的性能超过 1 200 DMIPS。

ARM Cortex-A8 微处理器是 ARM Cortex 新系列中第一款采用 ARMv7 架构中所有新技术的 ARM 处理器, 是 ARM 公司目前为止开发的速度最快、最节能的微处理器, 是灵活的低功耗、高性能处理领域的巨大跃迁; 最高能达到 2 000 MIPS, 使它成为运行多通道视频、音频和游戏应用的要求越来越高的消费产品的最佳选择。