



ARM处理器

与 **C** 语言开发应用

范书瑞 赵燕飞 高铁成 编著



北京航空航天大学出版社

TP332AR
330
1=

ARM 处理器与 C 语言 开发应用

范书瑞 赵燕飞 高铁成 编著

北京航空航天大学出版社

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。

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,赵燕飞,高铁成编著.

北京:北京航空航天大学出版社,2008.8

ISBN 978-7-81124-435-9

I. A… II. ①范…②赵…③高… III. ①微处理器,ARM—系统设计②C 语言—程序设计 IV. TP332 TP312

中国版本图书馆 CIP 数据核字(2008)第 107214 号

ARM 处理器与 C 语言开发应用

范书瑞 赵燕飞 高铁成 编著

责任编辑 李 杰 王淑梅

*

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(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:bhpress@263.net

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:20 字数:512 千字

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-435-9 定价:32.00 元

前言

ARM(Advanced RISC Machines)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:

文件夹内为各章节对应的编程实例,只包含本章节涉及到的程序代码。7.2文件夹内包含ADS环境下完整的开发代码。11文件夹内是针对S3C2410A开发的汇编初始化程序,简单实用。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。

编 者

2007年9月



第1章 ARM 微处理器概述

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	3
1.3 ARM 微处理器系列	4
1.3.1 ARM7 微处理器系列	5
1.3.2 ARM9 微处理器系列	5
1.3.3 ARM9E 微处理器系列	5
1.3.4 ARM10E 微处理器系列	6
1.3.5 ARM11 系列	6
1.3.6 Cortex 系列	6
1.3.7 SecurCore 系列	7
1.3.8 Intel 的 Xscale	7
1.3.9 Intel 的 StrongARM	8
1.4 应用领域和选型	8
1.4.1 应用领域	8
1.4.2 应用选型	9

第2章 ARM 编程模型和指令集介绍

2.1 ARM 编程模型	11
2.1.1 指令长度及数据类型	11
2.1.2 ARM 处理器的工作状态	11
2.1.3 ARM 体系结构的存储器格式	12
2.1.4 处理器模式	12
2.1.5 寄存器组织	13
2.1.6 程序状态寄存器	15

2.1.7 异常	16
2.2 ARM 处理器的指令格式	19
2.2.1 ARM 处理器的指令格式	19
2.2.2 指令的条件码	20
2.3 ARM 指令寻址方式	21
2.3.1 数据处理指令的操作数寻址方式	21
2.3.2 字及无符号字节的 Load/Store 指令的寻址方式	23
2.3.3 杂类 Load/Store 指令的寻址方式	24
2.3.4 批量 Load/Store 指令的寻址方式	25
2.3.5 协处理器 Load/Store 指令的寻址方式	27
2.4 ARM 指令集介绍	27
2.4.1 跳转指令	27
2.4.2 B 指令和 BL 指令	28
2.4.3 数据处理指令	28
2.4.4 程序状态寄存器访问指令	32
2.4.5 加载/存储指令	33
2.4.6 批量数据加载/存储指令	35
2.4.7 数据交换指令	36
2.4.8 协处理器指令	37
2.4.9 异常产生及 CLZ 指令	38
2.5 Thumb 指令及应用	39

2.6 DSP 扩展指令	39
2.6.1 带符号乘和乘加指令	39
2.6.2 饱和和整数运算	40
2.6.3 存储辅助指令	42

第 3 章 ARM 程序设计基础

3.1 ARM 汇编器所支持的伪操作	44
3.1.1 符号定义伪指令	44
3.1.2 数据定义伪指令	45
3.1.3 汇编控制伪指令	46
3.1.4 其他常用的伪指令	47
3.2 汇编语言程序设计	49
3.2.1 汇编语言程序中常用的符号	50
3.2.2 汇编语言程序中的表达式和运算符	51
3.2.3 汇编语言的程序结构	54
3.3 C 语言和汇编语言混合程序设计	55
3.3.1 汇编语言与 C/C++ 的混合编程	55
3.3.2 ATPCS 规则	59
3.4 编译器的使用	61
3.4.1 ARM 编译器 C/C++ 库	61
3.4.2 ARM 编译器关键词	62
3.4.3 ARM 编译器支持的数据类型	63
3.5 程序设计示例	64

第 4 章 S3C2410A 处理器硬件结构

4.1 S3C2410A 处理器概述	68
4.2 引脚介绍	70
4.3 特殊功能寄存器	76
4.4 最小应用系统设计	86
4.4.1 电源电路设计	86
4.4.2 晶振电路设计	87
4.4.3 复位电路设计	88
4.5 JTAG 调试接口设计	88
4.5.1 JTAG 概述	88

4.5.2 接口电路设计	90
4.5.3 访问处理器寄存器	91

第 5 章 存储系统设计

5.1 存储器控制器	92
5.1.1 存储器映像	92
5.1.2 接口信号时序	93
5.1.3 内存控制器	94
5.2 地址线接口设计	100
5.2.1 8 位存储器接口设计	100
5.2.2 16 位存储器接口设计	101
5.3 Nor Flash 接口设计	103
5.3.1 Nor Flash 和 Nand Flash 的区别	103
5.3.2 SST39VF1601 电路设计	105
5.4 Nand Flash 接口设计	106
5.4.1 性能	107
5.4.2 Nand Flash 控制功能寄存器	107
5.4.3 接口设计	110
5.5 SDRAM 接口电路设计	111

第 6 章 时钟和定时器

6.1 时钟和功耗管理	114
6.1.1 锁相环	114
6.1.2 时钟和功耗控制逻辑	115
6.1.3 时钟和功耗管理寄存器	117
6.1.4 程序设计	119
6.2 PWM 定时器	120
6.2.1 PWM 定时器操作	121
6.2.2 PWM 定时器控制寄存器	124
6.2.3 程序设计	127
6.3 实时时钟	128
6.3.1 RTC 功能描述	129
6.3.2 RTC 特殊功能寄存器	130
6.3.3 RTC 程序设计	133
6.4 看门狗	134
6.4.1 看门狗操作	135

6.4.2 看门狗寄存器	135	9.3.2 256 色调色板	198
6.4.3 看门狗程序设计	136	9.4 LCD 控制器的使用	200
第 7 章 通用端口和中断设计		9.4.1 控制器功能寄存器	200
7.1 I/O 端口控制寄存器	139	9.4.2 LCD 控制寄存器的初始化	206
7.1.1 端口寄存器	139	9.5 LCD 编程实例	206
7.1.2 外部中断寄存器	145	9.5.1 液晶屏初始化	206
7.1.3 通用状态寄存器	148	9.5.2 显示像素	209
7.1.4 端口程序设计	149	9.5.3 图形函数	209
7.2 S3C2410A 中断控制器	150	9.5.4 测试程序	213
7.2.1 中断控制	150	第 10 章 ADC 和触摸屏接口设计	
7.2.2 中断源	151	10.1 A/D 转换器	214
7.2.3 中断控制专用寄存器	153	10.1.1 A/D 转换步骤	214
7.2.4 中断子控制寄存器	155	10.1.2 ADC 的主要类型	216
7.2.5 外部中断程序设计	156	10.1.3 A/D 转换器的主要技术指标	223
第 8 章 串行接口设计		10.2 触摸屏原理	223
8.1 UART 接口及编程实例	159	10.2.1 触摸屏结构	224
8.1.1 UART 的操作	160	10.2.2 触摸屏的主要类型	224
8.1.2 UART 特殊功能寄存器	161	10.3 ADC 和触摸屏接口	226
8.1.3 编程实例	167	10.3.1 接口方式	226
8.2 IIC 总线接口及编程实例	173	10.3.2 功能描述	227
8.2.1 IIC 总线协议	173	10.3.3 ADC 和触摸屏专有寄存器	228
8.2.2 S3C2410A IIC 接口操作	174	10.4 ADC 和触摸屏接口编程实例	231
8.2.3 IIC 总线特殊功能寄存器	177	10.4.1 ADC 编程实例	231
8.2.4 编程实例	179	10.4.2 触摸屏编程实例	231
8.3 SPI 接口及编程实例	184	第 11 章 引导程序设计	
8.3.1 SPI 操作	184	11.1 引导机理分析	235
8.3.2 SPI 特殊功能寄存器	187	11.1.1 引导程序工作原理	235
8.3.3 编程实例	189	11.1.2 引导程序的启动过程	236
第 9 章 LCD 接口设计		11.2 ARM9 初始化代码分析	238
9.1 LCD 控制器的特性	191	11.2.1 外部文件	238
9.2 STN LCD 控制器基本操作	192	11.2.2 定义常量	241
9.2.1 视频操作	193	11.2.3 异常处理	242
9.2.2 显示驱动方式	194	11.2.4 主体程序	245
9.2.3 数据存储格式	195	11.2.5 调用 C 语言程序	249
9.3 TFT LCD 控制器基本操作	196	11.3 常用引导程序介绍	251
9.3.1 视频操作	197		

11.3.1 引导程序 VIVI	251	12.3.3 Linux 内核的移植	274
11.3.2 引导程序 U-Boot	253	12.3.4 根文件系统的实现	276
11.3.3 其他引导程序	255	12.4 应用程序的开发	279
第 12 章 嵌入式操作系统及其应用开发		12.4.1 Hello 测试程序	279
12.1 嵌入式操作系统概况	258	12.4.2 说话人识别算法的实现	279
12.1.1 嵌入式操作系统特点	258	12.5 网络编程实例	282
12.1.2 常用嵌入式操作系统	259	12.5.1 网络驱动程序	282
12.2 Linux 发展支柱和开发基础	261	12.5.2 socket 基本函数	284
12.2.1 Linux 发展支柱	261	12.5.3 TCP 编程实例	289
12.2.2 开发工具的使用	262	12.5.4 综合训练之 UDP 编程实现	292
12.3 嵌入式 Linux 的移植	268	附 录	295
12.3.1 建立交叉编译环境	268	参考文献	308
12.3.2 引导程序的移植	269		

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。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 授权市场里,ARM 占有 26% 的份额,远超过其他竞争对手。全球采用 ARM 技术的芯片一年总出货量超过了 20 个亿,预计到 2010 年总数将达 45 亿个。

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 7 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~v7。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。

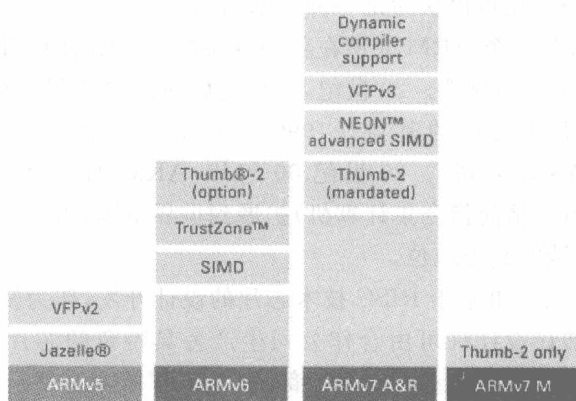


图 1.1 ARM 指令集的体系结构与扩展图

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×32 条或 128 位×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。

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 OptimoDE 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 微处理器系列

ARM 微处理器目前有 8 个系列可供选择,同时 Intel 也提供基于 ARM 处理器 Xscale 和

StrongARM 系列,每一个系列的 ARM 微处理器都有各自的特点和应用领域。

1.3.1 ARM7 微处理器系列

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器,适合用于对价位和功耗要求敏感的消费类型,包含 ARM720T、ARM7EJ-S、ARM7TDMI 和 ARM7TDMI-S 四种型号。

该系列处理器采用 3 级流水线 ARM 核,代码密度高并兼容 16 位的 Thumb 指令集,广泛支持包括 Windows CE、Symbian OS、Linux、Palm OS 等在内的操作系统。同时其指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容,便于用户的产品升级换代。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的需求。

ARM7 系列微处理器已广泛应用在工业控制、Internet 设备、网络和调制解调器设备、移动电话等多种多媒体和嵌入式应用领域。

1.3.2 ARM9 微处理器系列

ARM9 系列微处理器包含 ARM920T 和 ARM922T 两种类型,是以 ARM9TDMI 为原型发展而来的,其中 ARM920T 内置 8K 指令和数据 Cache;ARM922T 内置 18K 的双 Cache。二者均内置全性能的 MMU 和高速 AMBA 总线接口。AMBA 片上总线是一个开放标准,已成为 SOC 构建和 IP 库开发的事实标准。AMBA 先进的高性能总线 AHB 接口现已被所有新的 ARM 核支持,提供开发全综合设计系统。

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,支持 32 位 ARM 指令集和 16 位 Thumb 指令集,采用 5 级整数流水线,提供 1.1 MIPS/MHz 的哈佛结构,支持 32 位的高速 AMBA 总线接口。同时,提供全性能的内存管理单元(MMU),支持 Windows CE、Linux、Palm OS 等多种嵌入式操作系统。

ARM9 系列微处理器主要应用领域:

- 新一代手持设备,包括可视电话、便携通信设备、PDA;
- 数字消费类产品,包括机顶盒、家庭网关、游戏机、MPS、MPEG4;
- 图像处理,包括高端打印机、数字照相机和数字摄像机等。

1.3.3 ARM9E 微处理器系列

ARM9E 系列微处理器包含 ARM926EJ-S、ARM946E-S、ARM966E-S、ARM968E-S 和 ARM996HS 等 5 种类型。ARM9E 系列微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,极大地减少了芯片的面积和系统的复杂程度。ARM9E 系列微处理器在 ARM9 的基础上进一步支持 VFP9 浮点协处理器,主频最高可达 300 MIPS。ARM9E 系列微处理器提供了增强的 DSP 处理能力,很适合那些需要同时使用 DSP 和微控制器的应用场合。

1.3.4 ARM10E 微处理器系列

ARM10E 系列微处理器包含 ARM1020E、ARM1022E 和 ARM1026EJ - S 等类型。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 微处理器相比较,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先进的节能方式,使其功耗极低。

ARM10E 的核心在于采用 6 级整数流水线,指令执行效率更高,并使用向量浮点单元 VFP10 提供的高性能的浮点解决方案,从而极大地提高了处理器的整型和浮点运算性能,为用户提供图形引擎应用夯实基础。

1.3.5 ARM11 系列

ARM11 系列微处理器包括 ARM1136J(F)-S、ARM1156T2(F)-S 和 ARM1176JZ(F)-S 3 种类型。ARM11 处理器的最高速度可以达到 1 GHz 水平,与目前采用的 ARM9 和 ARM10 内核的处理器最高速度只能达到 400 MHz 的水平相比,ARM11 可谓取得了巨大的技术进步。

ARM11 着重提高了多媒体处理性能,它采用了 ARMv6 指令集技术,使得 ARM11 处理器在处理流媒体和 Java 程序时的性能表现更为出色。ARM11 具有更低的耗电量,可以满足下一代手持设备对电池供应时间的要求。ARM11 处理器支持目前得到广泛应用的各种嵌入式操作系统,如 Windows CE、Palm OS、Linux 以及 Symbian OS 等,而采用 ARM11 内核的嵌入式处理器也将是基于这些操作系统的手持设备的最佳选择。ARM11 处理器将不仅被用在 3G 手机以及 PDA 中,还将在其他要求紧凑高效的嵌入式设备中(如多媒体无线通信设备、家庭网关以及其他网络产品)一显身手,其强大的性能完全可以满足用户对语音和高速网络连接的需求。

1.3.6 Cortex 系列

Cortex 系列微处理器包括 Cortex - M3、Cortex - A8、Cortex - R4 和 Cortex - R4F 4 种类型。该系列采用 v7 结构,分为 R、A 和 M 3 种系列,分别针对不同的应用领域。

Cortex - M3 微处理器适用于高性能及低成本需求的嵌入式应用,如:微控制器、汽车系统、大型家用电器、网络装置等。与原来的通用型 ARM CPU 不同, Cortex - M3 主要针对单片机领域。Cortex - M3 采用 v7 指令集,它的速度比 ARM7 快 1/3,功耗低 3/4,并且能实现更小芯片面积,有利于将更多功能整合在更小的芯片尺寸中。Cortex - M3 是一个 32 位的核,在传统的单片机领域中,有一些不同于通用 32 位 CPU 应用的要求。Cortex - M3 完全基于硬件进行中断处理,最多可减少 12 个时钟周期数,在实际应用中可减少 70% 中断。针对单片机调试工具非常便宜的特点,Cortex - M3 采用了新型的单线调试技术,专门拿出一个引脚来做调试,从而节约了大笔的调试工具费用。同时,Cortex - M3 中还集成了大部分存储器和控制器,这样工程师可以直接在 MCU 外连接 Flash,从而降低了设计难度和应用障碍。ARM Cor-

Cortex-M3 处理器结合了多种突破性技术,可以使芯片供应商提供超低价格的芯片,仅 33 000 门的内核性能就可达 1.2 DMIPS/MHz。该处理器还集成了许多紧耦合系统外设,令系统能满足下一代产品的控制需求。

Cortex-A8 处理器最高性能达到 2 000 MIPS,使它成为对运行多通道视频、音频和游戏应用要求越来越高的消费产品的最佳选择,该处理器将给低功耗移动产品带来重大变革,使得最终用户可以享受到更高水准的娱乐和创新。由于在 65 nm 工艺下制造,使得 Cortex-A8 处理器的功耗不到 300 mW,第一次为低费用、高容量的产品带来了台式机级别的性能。

Cortex-R4 处理器可通过 MPU(Memory Protection Unit,内存保护单元)、高速缓存以及 TCM(Tightly Coupled Memory,紧密耦合内存)让处理器能针对各种不同的嵌入式应用进行最佳化调整,且不会影响基本的 ARM 指令集的兼容性,并可协助应用软件开发与 OEM 厂商重复运用现有的软件投资。Cortex-R4 处理器可支持新一代手机、硬盘、打印机及汽车电子设备,能协助新一代嵌入式产品快速执行各种复杂的控制算法与实时工作的运算。针对车用产品,Cortex-R4 处理器还在各种安全应用上加入容错功能和内存保护机制,支持最新版 OSEK 实时操作系统。对于针对引擎管理系统开发的系统单芯片组件而言,这是相当重要的特色,因为这种管理系统必须在许多外围组件中实时运作。

Cortex-R4F 处理器专门针对汽车市场的高级功能,包括:对纠错码(ECC)存储器的支持;错误侦测向相互连接的扩展;一个任意合成的浮点单元(FPU)。Cortex-R4F 处理器是建立在 Cortex-R4 处理器的先进功能基础之上的。这些功能包括在合成过程中的可配置性,以及通过高解析度的存储器保护单元、高速缓存存储器、高度搭配的存储器、DMA 和调试设备使不同应用的处理器达到最优化。

1.3.7 SecurCore 系列

SecurCore 系列微处理器包含 SecurCore SC100、SecurCore SC110、SecurCore SC200 和 SecurCore SC210 这 4 种类型,是专为安全需要而设计的,其提供了完善的 32 位 RISC 技术的安全解决方案。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,即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 体系结构的各种主要特点外,还在系统安全方面具有如下的特点:

- 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产品及系统,如电子商务、电子政务、电子银行业务、网络和认证系统等领域。

1.3.8 Intel 的 Xscale

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,已使用在数字移动电话、个人数字助理和网络产品等场合。Xscale 处理器是 Intel 目前主要推广的一款 ARM 微处理器。

1.3.9 Intel 的 StrongARM

Intel 的 StrongARM 处理器 SA-1100 是采用 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,采用在软件上兼容 ARMv4 的体系结构,同时采用具有 Intel 技术优点的体系结构。

Intel 的 StrongARM 处理器是便携式通信产品和消费类电子产品的理想选择,已成功应用于多家公司的掌上电脑系列产品。

1.4 应用领域和选型

1.4.1 应用领域

作为业界领先的 32 位嵌入式 RISC 微处理器,ARM 微处理器的应用几乎已经深入到各个领域:

1) 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。

2) 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,在该领域的地位日益巩固。

3) 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进行了优化,并获得广泛的支持,也对 DSP 的应用领域提出了挑战。

4) 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。

5) 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了 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许多不同的领域,并会在将来取得更加广泛的应用。ARM 公司对如此众多的应用领域进行了简单总结,并对 ARM 内核的选择做出指导:

1) 开放应用平台

包括无线系统、消费产品以及成像设备等,所选用内核为 ARM720T、ARM920T、ARM922T、ARM926EJ-S、ARM1020E、ARM1022E、ARM1026EJ-S、ARM11 MPCore、ARM1136J(F)-S、ARM1176JZ(F)-S 和 ARM Cortex-A8。

2) 实时嵌入式应用

包括存储、自动化、工业和网络设备,所选内核一般为 ARM7EJ - S、ARM7TDMI、ARM7TDMI - S、ARM946E - S、ARM966E - S、ARM968E - S、ARM996HS、ARM1026EJ - S、ARM1156T2(F) - S、ARM Cortex - M3、ARM Cortex - R4 和 ARM Cortex - R4F。

3) 安全系统

包括信用卡和 SIM 卡,所选内核为 SecurCore SC100 和 SecurCore SC200。

1.4.2 应用选型

鉴于 ARM 微处理器的众多优点,随着国内外嵌入式应用领域的逐步发展,ARM 微处理器必然会获得广泛的重视和应用。但是,由于 ARM 微处理器有多达十几种的内核结构,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一定的困难。以下从应用的角度出发,对在选择 ARM 微处理器时所应考虑的主要问题做一些简要的探讨。

1. ARM 微处理器内核的选择

如果希望使用 WinCE 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,存储管理单元)功能的 ARM 芯片,ARM720T、ARM920T、ARM922T、ARM946T 和 Strong - ARM 都带有 MMU 功能。ARM7TDMI 没有 MMU,不支持 Windows CE 和标准 Linux,但目前有 μ 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, μ Clinux 已经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表现。

2. 系统时钟控制器

系统时钟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。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 核或 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 芯片均需要根据各自不同的应用领域,扩展相关功能模块,并集成在芯片之中,称之为片内外围电路,如 USB 接口、IIS 接口、LCD 控制器、键盘接口、RTC、ADC、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,这样既可简化系统的设计,同时也能提高系统的可靠性。