

STM32[✈]系列

ARM Cortex-M3微控制器 原理与实践

王永虹 徐 炜 郝立平 编著



北京航空航天大学出版社



STM32[✈]系列

ARM Cortex-M3微控制器 原理与实践

本书介绍ARM Cortex-M3内核结构特点和Thumb-2指令集，及其与ARM其他内核的比较。详细阐述意法半导体（ST）公司STM32系列ARM Cortex-M3微控制器的编程模型、存储器结构、异常处理、电源管理、时钟与复位、嵌套向量中断控制器、调试单元，以及其他各种外设的结构和编程方法。说明STM32库函数的使用方法，并简要介绍STM32相应的开发环境、工具和应用实例。

本书所附的光盘中包含：IAR EWARM评估版软件；STM32-SK仿真评估板的USB仿真器驱动、示例程序；STM32F103-DK开发板的硬件资料、各种外设使用的示例程序；STM32的应用笔记（Application Note）中文版。

本书适用于使用STM32系列Cortex-M3微控制器的研发人员作为技术、编程参考，也可作为STM32系列微控制器教学或培训用教材。

上架建议：单片机/嵌入式系统

ISBN 978-7-81124-418-2



9 787811 244182 >

定价：49.00元（含光盘）

TP332.3/142D

2008

STM32 系列 ARM Cortex-M3 微控制器原理与实践

王永虹 徐 炜 郝立平 编著

微控(913)自解第5分通

本书为《微控制器原理与实践》(EM-32)的第5分通

微控(913)自解第5分通

7. 2008.12

8. 411 8212 87. 1/2008

微控(913)自解第5分通

1. 2008.12

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

微控(913)自解第5分通

北京航空航天大学出版社

(第1版) 2008.12 8212 87. 1/2008

新学网
PDG

内 容 简 介

本书介绍 ARM Cortex-M3 内核结构特点和 Thumb-2 指令集,及其与 ARM 其他内核的比较。详细阐述意法半导体(ST)公司 STM32 系列 ARM Cortex-M3 微控制器的编程模型、存储器结构、异常处理、电源管理、时钟与复位、嵌套向量中断控制器、调试单元,以及其他各种外设的结构和编程方法。说明 STM32 库函数的使用方法,并简要介绍 STM32 相应的开发环境、工具和应用实例。

本书所附的光盘中包含:IAR EWARM 评估版软件;STM32-SK 仿真评估板的 USB 仿真器驱动、示例程序;STM32F103-DK 开发板的硬件资料、各种外设使用的示例程序;STM32 的应用笔记(Application Note)中文版。

本书适用于使用 STM32 系列 Cortex-M3 微控制器的研发人员作为技术、编程参考,也可作为 STM32 系列微控制器教学或培训用教材。

图书在版编目(CIP)数据

STM32 系列 ARM Cortex-M3 微控制器原理与实践/王永虹,徐炜,郝立平编著. —北京:北京航空航天大学出版社,2008.7

ISBN 978-7-81124-418-2

I. S… II. ①王…②徐…③郝… III. 微控制器 IV. TP 332.3

中国版本图书馆 CIP 数据核字(2008)第 096077 号

© 2008,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本书及其所附光盘内容。

侵权必究。

STM32 系列 ARM Cortex-M3 微控制器原理与实践

王永虹 徐 炜 郝立平 编著

责任编辑 卫晓娜 王 艳

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

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:30.75 字数:787 千字

2008 年 7 月第 1 版 2008 年 7 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-418-2 定价:49.00 元(含光盘 1 张)

前言

ARM Cortex-M3 是 ARM 公司针对微控制器领域推出的新一代处理器架构,它采用 ARM v7-M 体系结构和面向高级语言的 Thumb-2 指令集,在代码密度、实时性、运算性能、功耗、价格等方面达到了很好的平衡。Cortex-M3 处理器不仅定义了传统意义上的处理器内核,也对存储器、时钟、复位、中断控制器、MPU、调试接口、电源管理等作了全面的规范,使采用 Cortex-M3 的各种芯片具有更统一的编程接口,简化了用户使用不同厂家芯片的复杂度。

意法半导体(ST)公司推出的基于 ARM Cortex-M3 内核的 STM32 系列微控制器,集 32 位 RISC 处理器、低功耗、高性能模拟技术、高速 DMA 通道及丰富的片内外设、JTAG 仿真调试等于一体,定义了新一代“超级单片机”的概念,加上丰富的技术资料 and 完善的开发工具,使用方便,具有极高的性价比。

在性能上,STM32 的 32 位 Cortex-M3 内核,时钟频率可达 72 MHz,指令速度可接近 80 MIPS。同时,它具有 32 位硬件除法和单周期乘法器等一系列先进的体系结构,大大增强了它的数据处理和运算能力,可以有效地实现一些数字信号处理的算法(如 FFT、DTMF 等)。在低功耗方面,其芯片功耗(3.3 V, 500 μ A/MIPS)已接近 MSP430 超低功耗单片机的水平。

在系统整合方面,STM32 系列单片机根据其不同产品,集成了多种功能模块,包括定时器、多功能串行接口(SPI/I²C/I²S/UART)、USB、CAN、12 位双路同步采样高速 ADC、看门狗定时器(WDT)、DMA 控制器、马达/电机控制器、多达 64 KB 的 RAM 和 256 KB Flash,以及丰富的中断功能。使用户可以根据应用需求,选择最合适的 STM32 系列产品来实现。

我们作为 ST 的合作伙伴,有机会第一时间接触、使用 STM32 系列微控制器。为更好地方便广大嵌入式开发人员推广、使用 STM32 系列微控制器,特组织、编写了这本书。这也是国内第一本介绍基于 ARM Cortex-M3 内核微控制器的中文书。由于书的篇幅所限,本书在最后定稿时,裁减了部分内容,主要是 STM32 的部分电气特性、开发工具使用方法和很多实验程序源代码。为保证内容的完整性和便于读者了解相关内容,把裁减的部分放入了本书所附的光盘中。为方便读者实践和开发,我们还设计开发了 STM32-SK、STM32-DK 等评估/开发板,配套的代码也都放

入了光盘。

本书由上海沁科信息技术有限公司和华东师范大学计算机系嵌入式系统实验室合作编写。其中第 2、3、4 章内容主要参考 ST 公司的 STM32 系列 Datasheet、Reference Manual、Software Library User Manual 等,最新的资料可以从 ST 的网站(mcu.st.com/mcu)下载。在此书的编写、审核过程中,沈建华、胡晓刚、黄文博、尤志坚、张敏、范皖勇、吴方锁、李吉、匡鑫、洪敦志、杨文博、杨蕾蕾、傅顺、项颖、朱海峰、徐峰、黄国富、吴家平等参与了资料整理、代码验证等工作;ST 公司技术经理梁平、市场经理 Cyrille Halbedel、Johnson Cao、技术工程师黄裕军、金尔雅等,在样片、资料、技术支持等方面做了很多工作;北航出版社胡晓柏,IAR 公司 Tony Ye 等也给予了很多关心和支持。在此,谨向他们表示衷心的感谢。

由于时间仓促及水平所限,以及 ST 的技术文档本身也在不断修正,错误及不妥之处在所难免,欢迎各位读者批评指正。有关本书和 STM32 芯片、开发工具的最新资讯,我们会在网站(www.mxchip.com)上及时发布,欢迎访问并相互交流。

编者

2008-5-25

目 录

第 1 章 ARM 及 Cortex-M3 处理器概述

1.1 ARM 处理器系列	1
1.1.1 命名规则	1
1.1.2 ARM 处理器系列	2
1.2 ARM Cortex-M3 处理器	5
1.2.1 处理器组件	5
1.2.2 Cortex-M3 的层次和实现选项	6
1.2.3 处理器内核	7
1.2.4 嵌套向量中断控制器(NVIC)	8
1.2.5 总线矩阵	8
1.2.6 集成调试	9
1.2.7 可选组件	9
1.2.8 Cortex-M3 处理器应用	9
1.3 ARM Cortex-M3 指令集	10
1.4 ARM Cortex-M3 的优势	18

第 2 章 STM32 系列微控制器

2.1 STM32 系列微控制器简介	22
2.1.1 STM32 微控制器的主要优点	22
2.1.2 STM32 微控制器的应用	24
2.2 STM32F101xx 系列微控制器	24
2.2.1 特 点	25
2.2.2 总体结构	26
2.3 STM32F103xx 系列微控制器	47
2.3.1 特 点	49
2.3.2 总体结构	50

第 3 章 STM32 系列微控制器存储器与外设

3.1 存储器和总线的结构	76
3.1.1 系统结构	76
3.1.2 存储器结构	77
3.1.3 存储器映射	77
3.1.4 启动配置	81

3.2 电源控制	82
3.2.1 电源供应	82
3.2.2 电源供应管理	83
3.2.3 低功耗模式	84
3.2.4 电源控制寄存器	88
3.3 复位和时钟控制	90
3.3.1 复位	90
3.3.2 时钟	91
3.3.3 RCC 寄存器描述	95
3.4 通用 I/O 和复用 I/O(GPIO 和 AFIO)	108
3.4.1 GPIO 功能描述	108
3.4.2 GPIO 寄存器描述	114
3.4.3 复用功能 I/O 和调试配置(AFIO)	117
3.4.4 AFIO 寄存器描述	121
3.5 中断和事件	126
3.5.1 嵌套向量中断控制器(NVIC)	126
3.5.2 外部中断/事件控制器(EXTI)	128
3.5.3 EXTI 寄存器	129
3.6 DMA 控制器	132
3.6.1 简介	132
3.6.2 主要特性	132
3.6.3 功能描述	133
3.6.4 DMA 寄存器	136
3.7 实时时钟(RTC)	141
3.7.1 简介	141
3.7.2 主要特性	141
3.7.3 功能描述	142
3.7.4 RTC 寄存器描述	144
3.8 备份寄存器(BKP)	149
3.8.1 简介	149
3.8.2 主要特性	149
3.8.3 干扰检测	150
3.8.4 RTC 校验	150
3.8.5 BKP 寄存器描述	150
3.9 独立的看门狗	153
3.9.1 简介	153
3.9.2 IWDG 寄存器描述	155
3.10 窗口看门狗(WWDG)	157
3.10.1 简介	157

3.10.2	主要特性	157
3.10.3	功能描述	157
3.10.4	如何编程看门狗的超时时间	158
3.10.5	调试模式	158
3.10.6	寄存器描述	159
3.11	高级控制定时器	161
3.11.1	简介	161
3.11.2	主要特性	161
3.11.3	框图	162
3.11.4	功能描述	162
3.11.5	TIM1 寄存器描述	192
3.12	通用定时器(TIMx)	209
3.12.1	简介	209
3.12.2	主要特性	209
3.12.3	框图	210
3.12.4	功能描述	211
3.12.5	TIMx 寄存器描述	238
3.13	控制器局域网(bxCAN)	252
3.13.1	简介	252
3.13.2	主要特性	252
3.13.3	总体描述	253
3.13.4	运行模式	254
3.13.5	功能描述	257
3.13.6	中断	265
3.13.7	寄存器访问保护	267
3.13.8	CAN 寄存器描述	267
3.14	内部集成电路(I ² C)接口	284
3.14.1	简介	284
3.14.2	主要特性	285
3.14.3	总体描述	286
3.14.4	功能描述	287
3.14.5	中断请求	296
3.14.6	I ² C 调试模式	297
3.14.7	I ² C 寄存器描述	297
3.15	串行外设接口(SPI)	305
3.15.1	简介	305
3.15.2	主要特性	305
3.15.3	功能描述	306
3.15.4	SPI 寄存器描述	313

3.16 通用同步异步收发机(USART)	318
3.16.1 简介	318
3.16.2 主要特性	318
3.16.3 总体描述	319
3.16.4 中断请求	337
3.16.5 USART 寄存器描述	338
3.17 USB 全速设备接口	345
3.17.1 概述	345
3.17.2 主要特性	346
3.17.3 结构框图	346
3.17.4 功能描述	347
3.17.5 编程中需要考虑的问题	348
3.17.6 USB 寄存器描述	356
3.18 模/数转换器(ADC)	366
3.18.1 概述	366
3.18.2 主要特性	367
3.18.3 引脚描述	367
3.18.4 功能描述	368
3.18.5 校准	372
3.18.6 数据对齐	373
3.18.7 基于通道的可编程的采样时间	373
3.18.8 外部触发转换	374
3.18.9 DMA 请求	374
3.18.10 双 ADC 模式	374
3.18.11 温度传感器	380
3.18.12 中断	381
3.18.13 ADC 寄存器描述	381
3.19 调试支持(DBG)	390
3.19.1 概述	390
3.19.2 相关的 ARM 文档	391
3.19.3 SWJ 调试端口(串行线和 JTAG)	392
3.19.4 引脚分布和调试端口引脚	392
3.19.5 STM32F10x JTAG TAP 连接	395
3.19.6 ID 编码和锁定机制	395
3.19.7 JTAG 调试端口	396
3.19.8 SW 调试端口	397
第 4 章 STM32 固件库	
4.1 STM32 固件库的定义规则	401
4.1.1 固件库命名规则	401

4.1.2 代码标准	402
4.2 STM32 库的层次结构	406
4.2.1 固件包描述	406
4.2.2 固件库文件描述	407
4.3 STM32 库的使用	408
第 5 章 STM32 系列微控制器开发工具与应用	
5.1 Keil MDK 介绍	412
5.1.1 开发过程及集成开发环境简介	412
5.1.2 工程管理	414
5.1.3 编写源程序	421
5.1.4 编译程序	425
5.1.5 调试程序	426
5.2 IAR EWARM 介绍	430
5.2.1 EWARM 集成开发环境及配套仿真器	431
5.2.2 在 EWARM 中生成一个新项目	434
5.2.3 编译和链接应用程序	439
5.2.4 用 J-LINK 调试应用程序	442
5.3 STM32-SK 仿真评估板	447
5.3.1 评估板规格说明	448
5.3.2 测试程序	451
5.3.3 关于仿真评估板的几个问题	456
5.4 STM32-DK 开发板	457
5.4.1 开发板规格说明	457
5.4.2 开发板实例程序	462
5.4.3 关于 ST ARM 的常见问题	466
5.5 mx-Pro 量产编程器使用简介	466
5.5.1 编程文件管理	467
5.5.2 芯片烧写	472
5.6 应用实例：基于 STM32 的数据采集器	473
5.6.1 硬件设计	473
5.6.2 软件设计	475
参考文献	481

ARM 及 Cortex-M3 处理器概述

ARM 在嵌入式应用领域取得了巨大的成功。从 1985 年的第一个 ARM1 原型诞生至今,已经有几十亿个 ARM 处理器被销售到了世界各地。ARM 的成功是建立在一个简单而又强大的原始设计之上的,随着技术的不断进步,这个设计也在不断的改进。ARM 内核、处理器并不是单一的,而是遵循相同设计理念、使用相似指令集架构的一个内核、处理器系列。

1.1 ARM 处理器系列

每个 ARM 处理器都有一个特定的指令集架构(ISA),而一个 ISA 版本又可以由多种处理器实现。ISA 随着嵌入式市场的需求而发展,至今已经有多个版本。ARM 公司规划该发展过程,使得在较早的架构版本上编写的代码也可以在后继版本上执行(即代码的兼容性)。

1.1.1 命名规则

早期 ARM 使用如图 1.1 所示的命名规则来描述一个处理器。在“ARM”后的字母和数字表明了一个处理器的功能特性。随着更多特性的增加,字母和数字的组合可能会改变。注意:命名规则不包含体系结构的版本信息。

关于 ARM 命名法则,还有一些附加的要点:

- ARM7TDMI 之后的所有 ARM 内核,即使“ARM”标志后没有包含那些字符,也都包括了 TDMI 功能特性。
- 处理器系列是共享相同硬件特性的一组处理器的具体实现。例如,ARM7TDMI、ARM740T 和 ARM720T 都共享相同的系列特性,都属于 ARM7 系列。
- JTAG 是由 IEEE1149.1 标准测试访问端口(Standard Test Access Port)和边界扫描结构来描述的。它是 ARM 用来发送和接收处理器内核与测试仪器之间调试信息的一系列协议。
- 嵌入式 ICE 宏单元(Embedded ICE macrocell)是建立在处理器内部用来设置断点和观察点的调试硬件。
- 可综合的意味着处理器内核是以源代码形式提供的,这种源代码形式又可以被编译成一种易于 EDA 工具使用的形式。

随着近年来 ARM 架构的产品爆炸性地涌入市场,以及对于维护架构一致性的高层次的要求,ARM 重新组织了 ARM 架构的规范,定义了以 ARM v7 架构的 Cortex 系列。

ARM {x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{-S}

x —— 系列

y —— 存储管理/保护单元

z —— cache

T —— Thumb 16位译码器

D —— JTAG调试器

M —— 快速乘法器

I —— 嵌入式跟踪宏单元

E —— 增强指令（基于TDMI）

J —— Jazelle

F —— 向量浮点单元

S —— 可综合版本

图 1.1 早期 ARM 命名规则

1.1.2 ARM 处理器系列

ARM 公司设计了许多处理器,它们可以根据使用的不同内核划分到各个系列中。系列划分是基于 ARM7、ARM9、ARM10、ARM11 和 Cortex 内核。后缀数字 7、9、10 和 11 表示不同的内核设计。数字的升序说明性能和复杂度的提高。ARM8 开发出来以后很快就被取代了。

在每个系列中,存储器管理、cache 和 TCM 处理器扩展也有多种变化。ARM 继续在可用的产品系列和每个系列内部的不同变种两方面做进一步开发。

表 1.1 总结了各种处理器的不同功能特性。值得注意的是,指令集架构(ISA)是体现 CPU 核性能特点的重要因素,如采用 v5TEJ 架构的 ARM926EJ-S 与采用 v4T 架构的 ARM920T 处理器,在相同的工作频率下,前者的处理能力要高得多。

表 1.1 ARM 处理器不同功能特性

CPU 核	MMU/MPU	Cache	Jazelle	Thumb	ISA 架构	E ^注
ARM7TDMI	无	无	否	是	v4T	否
ARM7EJ-S	无	无	是	是	v5TEJ	是
ARM720T	MMU	统一的 8 KB cache	否	是	v4T	否
ARM920T	MMU	独立的 16 KB/16KB D+I cache	否	是	v4T	否
ARM922T	MMU	独立的 8 KB/8 KB D+I cache	否	是	v4T	否
ARM926EJ-S	MMU	独立——cache 与 TCM 可配置	是	是	v5TEJ	是
ARM940T	MPU	独立的 4 KB/4 KB D+I cache	否	是	v4T	否
ARM946E-S	MPU	独立——cache 与 TCM 可配置	否	是	v5TE	是

续表 1.1

CPU 核	MMU/MPU	Cache	Jazelle	Thumb	ISA 架构	E ^注
ARM966E-S	无	独立——cache 与 TCM 可配置	否	是	v5TE	是
ARM1020E	MMU	独立的 32 KB/32 KB D+I cache	否	是	v5TE	是
ARM1022E	MMU	独立的 16 KB/16 KB D+I cache	否	是	v5TE	是
ARM1026EJ-S	MMU	独立——cache 与 TCM 可配置	是	是	v5TE	是
ARM1036J-S	MMU	独立——cache 与 TCM 可配置	是	是	v6	是
ARM1136JF-S	MMU	独立——cache 与 TCM 可配置	是	是	v6	是
Cortex-A8	MMU+TrustZone	独立可配置	是	是	v7	是
Cortex-M3	MPU 可选	无	否	是	v7	否
Cortex-R4	MPU	可选 TCM 可配置	否	是	v7	是

注：E 扩展提供了增强的乘法和饱和运算指令(DSP)。

1. ARM7 系列

ARM7 内核是冯·诺伊曼体系结构,数据和指令使用同一条总线。内核有一条 3 级流水线,执行 ARMv4 指令集。

ARM7TDMI 是 ARM 公司于 1995 年推出的新系列中的第一个处理器内核,是目前一个非常流行的内核,已被用在许多 32 位嵌入式处理器上。它提供了非常好的功耗比。ARM7TDMI 处理器内核已经许可给许多世界顶级半导体公司,它是第一个包括 Thumb 指令集、快速乘法指令和嵌入式 ICE 调试技术的内核。

ARM7 系列中一个重要的变化是 ARM7TDMI-S。ARM7TDMI-S 与标准 ARM7TDMI 有相同的操作特性,但它是可综合的。

ARM720T 是 ARM7 系列中最具灵活性的成员,因为它包含了一个 MMU。MMU 的存在意味着 ARM720T 能够处理 Linux 和 Microsoft 嵌入式操作系统(如 WinCE)。这一处理器还包括了一个 8 KB 的统一 cache(指令/数据混合 cache)。向量表可以通过设置一个协处理器 15(CP15)寄存器来重定位到更高的地址。

另一个成员是 ARM7EJ-S 处理器,它也是可综合的。ARM7EJ-S 与其他 ARM7 处理器有很大不同,因为它有一条 5 级流水线,并且执行 ARMv5TEJ 指令。这个版本是 ARM7 中唯一提供 Java 加速和增强指令,而没有任何存储器保护的处理器。

2. ARM9 系列

ARM9 系列于 1997 年问世。由于采用了 5 级指令流水线,ARM9 处理器能够运行在比 ARM7 更高的时钟频率上,提高了处理器的整体性能。存储器系统根据哈佛体系结构重新设计,区分了数据 D 和指令 I 总线。

ARM9 系列的第一个处理器是 ARM920T,它包含独立的 D+I cache 和一个 MMU。这个处理器能够被用在要求有虚拟存储器(虚存)支持的操作系统上。ARM922T 是 ARM920T 的变种,只有一半大小的 D+I cache。

ARM940T 包括一个更小的 D+I cache 和一个 MPU。它是针对不要求运行平台操作系统的应用而设计的。ARM920T 和 ARM940T 都执行 v4T 架构指令。

ARM9 系列的下一个处理器是基于 ARM9E-S 内核的。这个内核是 ARM9 内核带有 E 扩展的一个可综合版本。它有两个变种:ARM946E-S 和 ARM966E-S。两者都执行 v5TE 架构指令。它们也支持可选的嵌入式跟踪宏单元(ETM),它允许开发者实时跟踪处理器上指令和数据的执行。当调试对时间敏感(time-critical)的程序段时,这种方法非常重要。

ARM946E-S 包括 TCM、cache 和一个 MPU。TCM 和 cache 的大小可配置。该处理器是针对要求有确定的实时响应的嵌入式控制应用而设计的。而 ARM966E 有可配置的 TCM,但没有 MPU 和 cache 扩展。

ARM9 产品线的最新内核是 ARM926EJ-S 可综合的处理器内核,发布于 2000 年。它是针对小型便携式 Java 设备,诸如 3G 手机和个人数字助理(PDA)应用而设计的。ARM926EJ-S 是第一个包含 Jazelle 技术(可加速 Java 字节码的执行)的 ARM 处理器内核。它还有一个 MMU、可配置的 TCM,以及具有零或非零等待存储器的 D+I cache。

3. ARM10 系列

ARM10 发布于 1999 年,主要是针对高性能的设计。它把 ARM9 的流水线扩展到 6 级,也支持可选的向量浮点单元(VFP),它对 ARM10 的流水线加入了第 7 段。VFP 明显提高了浮点运算的性能,并与 IEEE754.1985 浮点标准兼容。

ARM1020E 是第一个使用 ARM10E 内核的处理器。像 ARM9E 一样,它包括了增强的 E 指令。它有独立的 32 KB D+I cache、可选向量浮点单元(VFP),以及 MMU。ARM1020E 还有一个双 64 位总线接口以提高性能。

ARM1026EJ-S 非常类似于 ARM926EJ-S,但同时具有 MPU 和 MMU。这一处理器具有 ARM10 的性能和 ARM926EJ-S 的灵活性。

4. ARM11 系列

ARM1136J-S 发布于 2003 年,是针对高性能和高能效应用而设计的。ARM1136J-S 是第一个执行 ARMv6 架构指令的处理器。它集成了一条具有独立的 load-store 和算术流水线的 8 级流水线。ARMv6 指令包含了针对媒体处理的单指令流多数据流(SIMD)扩展,特殊的设计以提高视频处理性能。

ARM1136JF-S 就是为了进行快速浮点运算,而在 ARM1136J-S 增加了向量浮点单元。

5. ARM Cortex 系列

ARM Cortex 发布于 2005 年,为各种不同性能需求的应用提供了一整套完整的优化解决方案,该系列的技术划分完全针对不同的市场应用和性能需求。目前 ARM Cortex 定义了以下三个系列。

Cortex-A 系列:针对复杂 OS 和应用程序(如多媒体)的应用处理器。支持 ARM、Thumb 和 Thumb-2 指令集,强调高性能与合理的功耗,存储器管理支持虚拟地址。

Cortex-R 系列:针对实时系统的嵌入式处理器。支持 ARM、Thumb 和 Thumb-2 指令集,强调实时性,存储器管理只支持物理地址。

Cortex-M 系列:针对价格敏感应用领域的嵌入式处理器。只支持 Thumb-2 指令集,强调操作的确定性,以及性能、功耗和价格的平衡。

这些系列曾在 ARMv7 发展过程中被正式介绍过,A 系列和 R 系列就已经隐式地出现在早期的版本中了,以及虚拟存储系统架构(VMSA)和保护存储系统架构(PMSA)。

到目前为止,Cortex 系列正式发布的版本为 Cortex-A8、Cortex-R4 和 Cortex-M3,它们全部

实现了 Thumb-2 指令集(或子集),可满足不同的性能、价格市场需求。

1.2 ARM Cortex-M3 处理器

Cortex-M3 处理器是一个低功耗的处理器,具有门数少,中断延迟小,调试容易等特点。它是为功耗和价格敏感的应用领域而专门设计的、具有较高性能的处理器,应用范围可从低端微控制器到复杂 SoC。

Cortex-M3 处理器使用了 ARM v7-M 体系结构,是一个可综合的、高度可配置的处理器。它包含了一个高效的哈佛结构三级流水线,可提供 1.25 DMIPS/MHz 的性能。在一个具有 32 个物理中断的标准处理器上实现(0.13 μm Metro @50 MHz),达到了突出的 0.06 mW/MHz 能效比。

为降低器件成本,Cortex-M3 处理器采用了与系统部件紧耦合的实现方法,来缩小芯片面积,其内核面积比现有的三级流水线内核缩小了 30%。Cortex-M3 处理器实现了 Thumb-2 指令集架构,具有很高的代码密度,可降低存储器需求,并能达到非常接近 32 位 ARM 指令集的性能。

对于系统和软件开发,Cortex-M3 处理器具有以下优势:

- 小的处理器内核、系统和存储器,可降低器件成本。
- 完整的电源管理,很低的功耗。
- 突出的处理器性能,可满足挑战性的应用需求。
- 快速的中断处理,满足高速、临界的控制应用。
- 可选的存储器保护单元(MPU),提供平台级的安全性。
- 增强的系统调试功能,可加快开发进程。
- 没有汇编代码要求,简化系统开发。
- 宽广的适用范围:从超低成本微控制器到高性能 SoC。

Cortex-M3 处理器在高性能内核基础上,集成了多种系统外设,可以满足不同应用对成本和性能的要求。处理器是全部可综合、高度可定制的(包括物理中断、系统调试等),Cortex-M3 还有一个可选的细粒度的(fine-granularity)存储器保护单元(MPU)和一个嵌入式跟踪宏单元(ETM)。

1.2.1 处理器组件

Cortex-M3 组件图如图 1.2 所示。

注意: ETM 和 MPU 是可选组件,在某些实现中可能不存在。

Cortex-M3 处理器主要包括:

- 处理器内核。
- 与处理器核紧密结合的嵌套向量中断控制器(NVIC)以实现低延迟的中断处理。
- 存储器保护单元(MPU),可选部件 MPU 实现存储器保护。
- 总线接口。
- 调试接口。

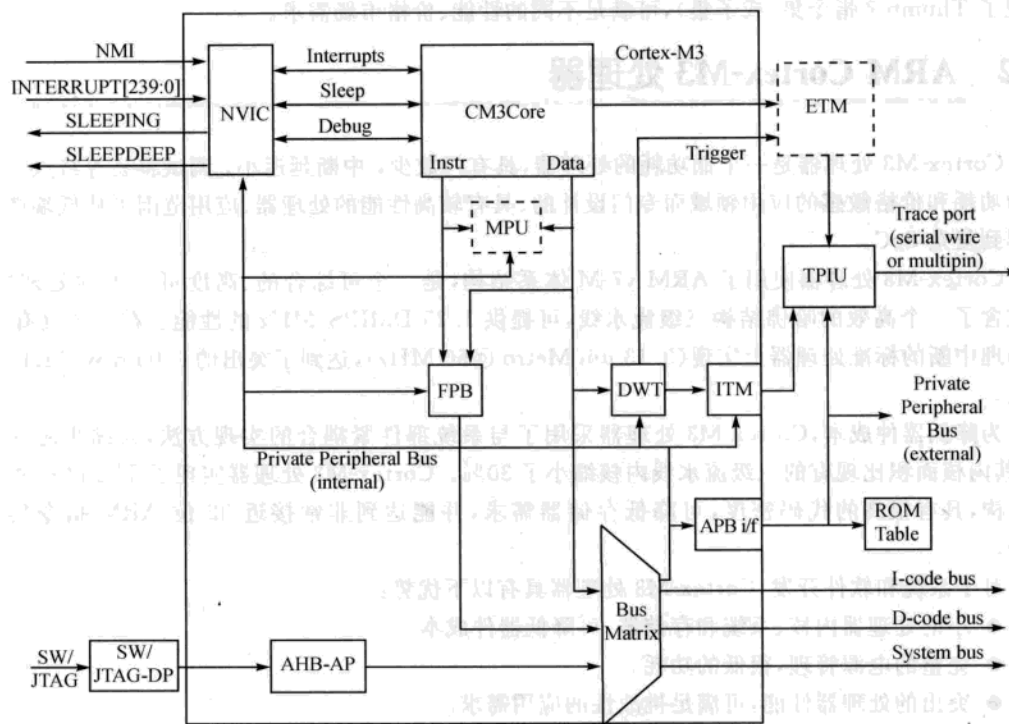


图 1.2 Cortex-M3 组件图

1.2.2 Cortex-M3 的层次和实现选项

处理器组件有两个层次,如图 1.2 所示,这是 RTL 设计层次。ETM、TPIU、SW/JTAG-DP 和 ROM 表 4 个组件在 Cortex-M3 层的外部,因为这些组件要么是可选的,要么就是在实现和使用时可以灵活改变的。实际的实现可能与图 1.2 不一样,可能的实现选项如下:

1. TPIU

TPIU 是 ITM、ETM(如果存在)和片外跟踪端口分析器之间传输 Cortex-M3 跟踪数据的桥梁。TPIU 可以设置成支持低开销调试的串行引脚跟踪,或者用于更高带宽跟踪的多引脚跟踪。TPIU 是与 CoreSight 兼容的。

TPIU 的实现选项如下:

- 如果系统中有 ETM 组件,那么就有 TPIU 格式器,否则就没有。
- 多个 TPIU 中的一个就可以跟踪多核的实现。
- ARM TPIU 部件可以被特定的 CoreSight 相应的 TPIU 替代。
- 在一个实际器件中,可能没有 TPIU。

注意:如果没有 TPIU,那么 Cortex-M3 就没有跟踪能力。

2. SW/JTAG-DP

处理器可以设置成有 SW-DP 或者 JTAG-DP 调试端口,或者两者都有。调试端口提供对系统中所有的外设寄存器、存储器、处理器寄存器的调试访问。