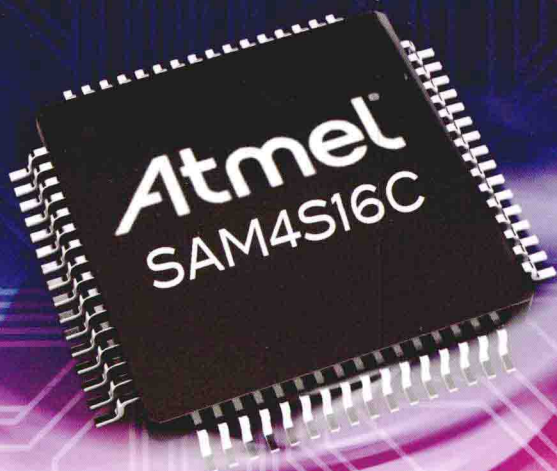


Atmel 中国大学计划教材

嵌入式系统 应用开发教程

——基于SAM4S

高新岩 朱明 于成 马艳华 马洪连 王亚维 主编 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

Atmel 中国大学计划教材

嵌入式系统应用开发教程

——基于 SAM4S

马洪连 主编
高新岩 朱明 于成 马艳华 编著



北京航空航天大学出版社

TP332

342

内 容 简 介

本书从实用角度出发,主要介绍 Atmel 32 位 SAM4S16C 微控制器及基于 SAM4S - EK 系统平台的应用开发。SAM4S16C 微控制器及 SAM4S - EK 应用平台由 Atmel 公司推出,具有高效信号处理功能,以及低功耗、低成本和易于使用的优点,是可满足专门面向电动机控制、汽车、电源管理、工业自动化及物联网等方面要求的灵活解决方案。

书中首先对 SAM4S16C 微控制器体系结构、工作原理、设计方法和相关的开发环境作了系统的介绍;接着介绍了 Atmel 公司最新推广的 SAM4S16C - EK 嵌入式系统应用开发平台的结构组成和功能;然后介绍了基于该系统开发平台操作系统的移植和应用;最后介绍了基于 SAM4S 微控制器系统的设计与应用开发实例。

本书结构合理、实例丰富,具有很强的实践性和实用性,可供高等学校计算机应用、电子信息工程、自动化、机电一体化等相关专业作为教材或参考书使用,也适合从事嵌入式系统开发的工程设计人员和广大嵌入式系统设计与开发的爱好者使用。

图书在版编目(CIP)数据

嵌入式系统应用开发教程:基于 SAM4S / 马洪连主编. -- 北京:北京航空航天大学出版社, 2015. 1
ISBN 978 - 7 - 5124 - 1653 - 6

I. ①嵌… II. ①马… III. ①微处理器—系统开发—教材 IV. ①TP332

中国版本图书馆 CIP 数据核字(2014)第 281688 号

版权所有,侵权必究。

嵌入式系统应用开发教程 ——基于 SAM4S

马洪连 主编

高新岩 朱明 于成 马艳华 王亚维 编著

责任编辑 杨 昕

*

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

北京楠海印刷厂印装 各地书店经销

*

开本:710×1 000 1/16 印张:28.25 字数:602 千字

2015 年 1 月第 1 版 2015 年 1 月第 1 次印刷 印数:3 000 册

ISBN 978 - 7 - 5124 - 1653 - 6 定价:59.00 元

前 言

随着嵌入式系统应用的普及,对嵌入式系统设计的技术人才的需求越来越大,与此同时也迫切需要有一些针对性更强的、适用于不同层次人员使用的教材和参考书。本书主要面向从事 32 位微控制器系统开发和系统应用设计的技术人员。

本书是应 Atmel 公司的邀请,编写基于该公司全球最新推广的 SAM4S-EK 嵌入式系统应用平台的相应理论教学与实践操作的参考教材。Atmel 公司的产品在国内影响较大,该平台已经在国内大学计划中得到了广泛的使用。

本书从实用的角度出发,主要介绍 Atmel 32 位 SAM4S16C 微控制器及基于 SAM4S-EK 系统平台的应用开发等内容。SAM4S16C 微控制器和 SAM4S-EK 平台是由 Atmel 公司推出的,具有高效信号处理功能,以及低功耗、低成本和易于使用的优点,是可满足专门面向电动机控制、汽车、电源管理、工业自动化及物联网等方面要求的灵活解决方案。

全书共分为 7 章:第 1 章主要介绍 Cortex-M4 处理器核的基本知识;第 2 章介绍 SAM4S-EK 应用平台的调试与开发;第 3 章介绍 SAM4S 系列微控制器结构组成与部件功能;第 4 章介绍 SAM4S-EK 系统平台的结构组成;第 5 章介绍 SAM4S-EK 开发平台接口及应用;第 6 章介绍嵌入式实时操作系统及操作系统的移植;第 7 章介绍 SAM4S 微处理器的综合设计与应用实例,帮助读者了解如何进行基于 SAM4S 微控制器系统的设计与应用开发。

本书主要介绍嵌入式微控制器的开发与应用技术,并配合 Atmel 公司的开发应用平台,力图满足高等院校、中小企业、嵌入式爱好者等多方群体的需求。书中作者既有教学经验丰富的教授,又有常年进行开发的技术人员;内容上既有严密的理论知识论述,又包含实用的嵌入式开发经验和成果总结,因此是一本不可多得的实用教程和手边工具参考书。

在本书的编写过程中得到了多方面的支持和帮助。首先得到了 Atmel 公司的大力支持,Atmel 公司为作者提供了 SAM4S-EK 开发平台和相关资料,Atmel 公司中国区 ARM MCU/eMPU 产品线总监庞长富博士就本书的内容、章节安排提出了非常有价值的建议。大连悠龙软件科技有限公司在技术上也给作者提供了大量无私



的帮助。另外,作者还要感谢北京航空航天大学出版社的编辑,是他们的大力支持才使本书能尽快地出版并发行。该书在编写过程中参考和引用了大量文献、资料和书刊,在此对参考文献中所有的作者深表谢意。互联网也是本书的一个参考来源,由于网上许多资料无法找到出处,所以如有内容涉及相关人士的知识产权,请给予谅解并及时与我们联系。

由于本书作者经验与水平的限制,书中如出现疏漏或不适宜的内容,希望读者给予批评指正,在此表示感谢。

作者的电子邮件地址:mhl@dlut.edu.cn。

编者

2014年8月

目 录

第 1 章 Cortex - M4 处理器核简介	1
1.1 概 述	1
1.2 Cortex - M4 总体组织结构	3
1.3 Cortex - M4 寄存器组织	6
1.4 Cortex - M4 工作模式及中断异常处理	10
1.4.1 工作模式	10
1.4.2 中断与异常处理	11
1.5 数据类型和存储格式	17
1.6 存储保护单元编程模型	18
1.6.1 MPU 概述	18
1.6.2 MPU 设置与使用	23
1.6.3 MPU 访问权限特性	23
1.7 浮点运算单元	25
1.8 Cortex - M4 指令集	30
1.8.1 Thumb 指令集	30
1.8.2 CMSIS 扩展指令集	36
第 2 章 调试系统与开发工具	38
2.1 Cortex - M4 调试系统结构	38
2.1.1 CoreSight 调试体系结构	38
2.1.2 Cortex - M4 调试结构	40
2.2 SAM - ICE 开发调试器	44
2.2.1 JTAG 边界扫描调试端口	44
2.2.2 SAM - ICE 调试器简介	44
2.2.3 微控制器的调试	46
2.3 常用的微控制器开发软件	48
2.4 Atmel Studio 专用开发环境	50
2.4.1 Atmel Studio 简介	50



2.4.2	Atmel Studio 的安装	50
2.4.3	新工程的建立	54
2.5	其他常用的开发工具和开发环境简介	55
第 3 章	SAM4S 系列微控制器	56
3.1	SAM4S 微控制器概述	56
3.2	SAM4S16C 微控制器内部总体结构	58
3.2.1	概 述	60
3.2.2	存储器组织与地址映射	66
3.2.3	启动机制	70
3.3	供电控制器 SUPC	71
3.3.1	SAM4S 微控制器电源供给	71
3.3.2	SUPC 结构组成	73
3.3.3	工作原理	75
3.3.4	应用程序设计	81
3.4	复位控制器 RSTC	84
3.4.1	RSTC 结构组成	85
3.4.2	工作原理	85
3.4.3	应用程序设计	91
3.5	实时时钟 RTC	95
3.5.1	RTC 结构组成	95
3.5.2	工作原理	97
3.6	实时定时器 RTT	98
3.6.1	RTT 结构组成	98
3.6.2	工作原理	99
3.6.3	应用程序设计	100
3.7	看门狗定时器 WDT	106
3.7.1	WDT 结构组成	106
3.7.2	工作原理	107
3.8	嵌套向量中断控制器 NVIC	108
3.8.1	NVIC 结构组成	108
3.8.2	工作原理	109
3.8.3	应用程序设计	111
3.9	外设 DMA 控制器 PDC	118
3.9.1	PDC 结构组成	118
3.9.2	工作原理	119

3.9.3 应用程序设计	121
3.10 通用并行输入输出接口 GPIO	124
3.10.1 GPIO 结构组成	124
3.10.2 工作原理	128
3.10.3 应用程序设计	141
第 4 章 SAM4S - EK 系统应用开发平台	143
4.1 SAM4S - EK 应用开发平台架构简介	143
4.2 平台组件与接口	147
4.3 快速 Flash 编程接口与应用	155
4.3.1 快速 Flash 结构组成	155
4.3.2 并行快速 Flash 编程	158
4.3.3 应用程序设计	163
4.4 增强内嵌 Flash 控制器 EEFC	165
4.4.1 EEFC 结构组成	165
4.4.2 工作原理	166
4.4.3 应用程序设计	174
4.5 静态存储控制器 SMC	183
4.5.1 SMC 结构组成	183
4.5.2 工作原理	185
4.6 LCD 背光控制和触摸屏	200
4.6.1 LCD 背光控制和触摸屏结构组成	200
4.6.2 工作原理	203
4.6.3 应用程序设计	205
4.7 通用异步收发器 UART	208
4.7.1 UART 结构组成	208
4.7.2 工作原理	209
4.7.3 应用程序设计	213
4.8 通用同步/异步收发器 USART	216
4.8.1 USART 结构组成	216
4.8.2 工作原理	218
4.8.3 应用程序设计	232
第 5 章 SAM4S - EK 开发平台接口与应用	241
5.1 脉宽调制控制器 PWM	241
5.1.1 PWM 结构组成	241



5.1.2	工作原理	245
5.1.3	应用程序设计	258
5.2	模/数转换器 ADC	262
5.2.1	ADC 结构组成	262
5.2.2	工作原理	264
5.2.3	应用程序设计	270
5.3	数/模转换器 DAC	282
5.3.1	DAC 结构组成	282
5.3.2	工作原理	283
5.3.3	应用程序设计	285
5.4	串行外设接口 SPI	294
5.4.1	SPI 结构组成	294
5.4.2	工作原理	295
5.4.3	应用程序设计	304
5.5	双总线接口 TWI	312
5.5.1	TWI 结构组成	312
5.5.2	工作原理	314
5.5.3	应用程序设计	326
5.6	同步串行控制器 SSC	330
5.6.1	SSC 结构组成	330
5.6.2	工作原理	332
5.6.3	应用程序设计	336
5.7	高速 USB 设备接口 HSUDP	341
5.7.1	HSUDP 结构组成	341
5.7.2	工作原理	343
5.7.3	应用程序设计	351
5.8	高速多媒体卡接口 HSMCI	356
5.8.1	HSMCI 结构组成	356
5.8.2	工作原理	358
5.8.3	应用程序设计	364
第 6 章 嵌入式实时操作系统及操作系统的移植		373
6.1	嵌入式实时操作系统概述	373
6.2	$\mu\text{C}/\text{OS}-\text{II}$ 及其在 AT91SAM4S-EK 平台上的移植	382
6.2.1	$\mu\text{C}/\text{OS}-\text{II}$ 实时操作系统简介	382
6.2.2	$\mu\text{C}/\text{OS}-\text{II}$ 操作系统的移植	384

6.3 FreeRTOS 及其在 AT91SAM4S-EK 平台上的移植	396
6.3.1 FreeRTOS 简介	396
6.3.2 FreeRTOS 操作系统的移植	397
6.4 μ Tenux 及其在 AT91SAM4S-EK 平台上的移植	407
6.4.1 μ Tenux 实时操作系统简介	407
6.4.2 μ Tenux 系统的移植	407
第 7 章 设计案例与开发实践	415
7.1 嵌入式应用系统设计概述	415
7.2 嵌入式微控制器综合实验教学平台设计实例	419
7.2.1 概 述	419
7.2.2 系统总体设计	420
7.2.3 系统硬件电路设计与实现	421
7.2.4 系统软件部分设计与实现	427
7.2.5 系统测试	437
参考文献	441

第 1 章

Cortex - M4 处理器核简介

1.1 概 述

Atmel 公司拥有多种基于 ARM 核的 32 位处理器,以满足用户的不同需求。同时也一直致力于基于 ARM 处理器的微控制器(MCU)产品开发,最近 Atmel 公司公布了第五代基于 Cortex - M4 的快闪微控制器。到 2012 年为止,Atmel 公司的 SAM3 和 SAM4 系列产品使 Cortex - M 系列产品组合的数目增加至 4 倍,ARM 系列的微控制器达到 200 多种。其中,包括高达 2 MB 的片上快闪存储器、192 KB SRAM 和包含高速 USB 在内的多种外设器件。2011 年发布的 Cortex - M4 系列的数款器件中还增加带有浮点运算单元(FPU),从而使 Atmel 公司基于 ARM 处理器产品的范围扩大到数字信号控制器市场。

Cortex - M4 是基于高性能的 32 位处理器核进行设计的,本身具有功耗低,门数少,中断延迟短,调试成本低等优点,为开发人员提供了极大的便利。Cortex - M4 包括快速中断处理性能,通过提高断点和跟踪能力来增强系统调试功能,高效的处理器内核、系统和内存,超低功耗的睡眠模式和集成的内存保护单元的安全平台。Cortex - M4 结构框图,如图 1 - 1 所示。

在 Cortex - M4 处理器核中,采用了具有三级流水线的哈佛结构,因此非常适合要求苛刻的嵌入式应用。该处理器高效的指令集和优化的设计提高了电源效率,并提供了各种单周期、SIMD 乘法、乘法与累加功能、饱和算法和专用的硬件除法等功能。

为促进低成本敏感型设备的设计,Cortex - M4 处理器核采用紧耦合系统部件以减小处理器尺寸,改善中断处理和系统调试能力,同时实现了代码高度密集以降低程序存储器的要求。

Cortex - M4 的指令集提供 32 位架构,具有 8 位和 16 位的高代码密度形式,还支持 Thumb/Thumb - 2 指令集,其中所采用的 Thumb - 2 指令集具有更高的指令效率和更强的性能。Thumb - 2 指令集结合了 16 位指令的代码密度和 32 位指令的性能,其底层关键特性使得 C 代码的执行变得更加自然。

Cortex - M4 系列处理器核采用了 CoreSight 调试跟踪体系结构,支持 8 个断点和 4 个数据观察点。在支持传统的 JTAG 基础上,还支持更好的低成本串行线调试

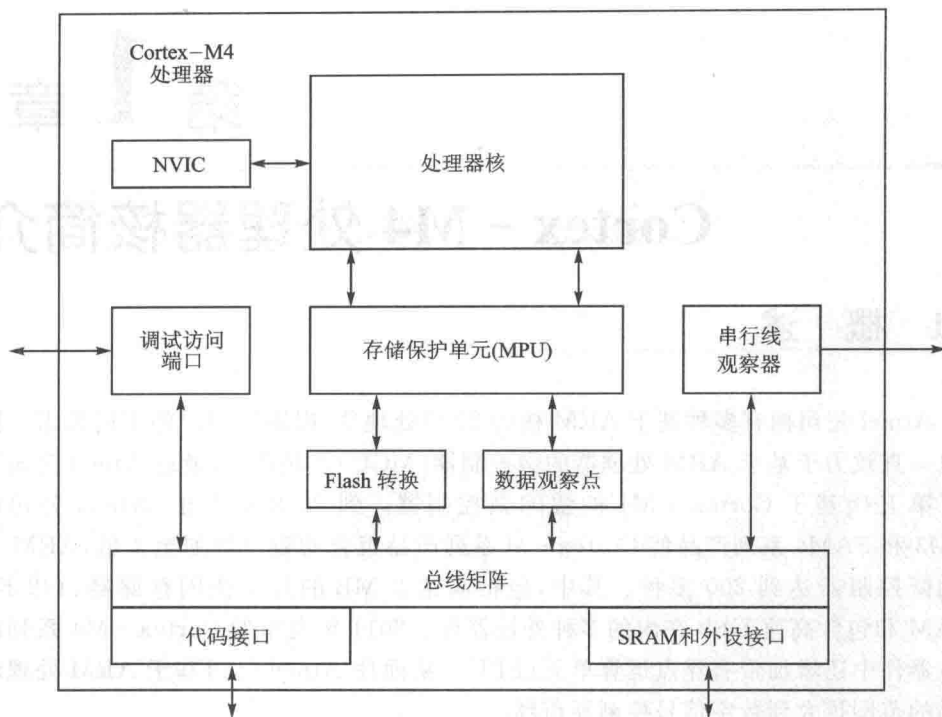


图 1-1 Cortex-M4 处理器核结构框图

接口(Single Wire,简称 SW)。处理器核内部的数据观测与跟踪单元(Data Watchpoint and Trace,简称 DWT)、测量跟踪宏单元(Instrumentation Trace Macrocell,简称 ITM)和可选的嵌入式跟踪宏单元(Embedded Trace Macrocell,简称 ETM)能获得处理器的指令跟踪流,提供低成本的实时跟踪能力。

Cortex-M4 处理器核与 ARM7、ARM9 处理器存在以下显著不同之处:

- ① 采用 ARMv7-M 体系结构;
- ② 不支持 ARM 指令集,仅支持 Thumb/Thumb-2 指令集;
- ③ 没有 Cache,也没有 MMU;
- ④ 具有 SW 跟踪调试接口;
- ⑤ 中断控制器内建于 Cortex-M4 之中;
- ⑥ 向量表内容为地址,而非指令;
- ⑦ 中断时自动保存和恢复状态,不支持协处理器。

在 Cortex-M4F 系列中,还提供了浮点运算单元 FPU。采用 32 位指令单精度(C 语言的 float)数据处理操作,硬件支持转换、加法、减法、乘法以及可选的累加、除法和平方根,同时硬件支持非格式化方式和所有的 IEEE 舍入模式。内部具有 32 个专用的 32 位单精度寄存器,可寻址用作 16 位双字寄存器。

1.2 Cortex - M4 总体组织结构

Cortex - M4 处理器内部基本结构主要包括 Cortex - M4 核、嵌套矢量中断控制器 NVIC、总线阵列 Bus Matrix、Flash 转换及断点单元 FPB、数据观测和跟踪单元 DWT、测量跟踪宏单元 ITM、存储器保护单元 MPU、嵌入式跟踪宏单元 ETM、跟踪接口单元 TPIU、存储器表 ROM Table、串行线调试接口 SW/SWJ - DP 等模块,其中 MPU 和 ETM 单元是可选单元。本节将分别对其主要模块进行介绍,以帮助读者了解 Cortex - M4 的基本结构。Cortex - M4 内部结构如图 1 - 2 所示。

以上这些单元可以分为内外两个层次,其中 ETM、TPIU、ROM Table、SW/SWJ - DP 和 WIC 单元属于外层。因为这几个单元或可选或可灵活配置实现,也就是在处理器具体实现时,TPIU、ROM Table、SW/SWJ - DP 和 WIC 可能与图 1 - 2 所示的不同。

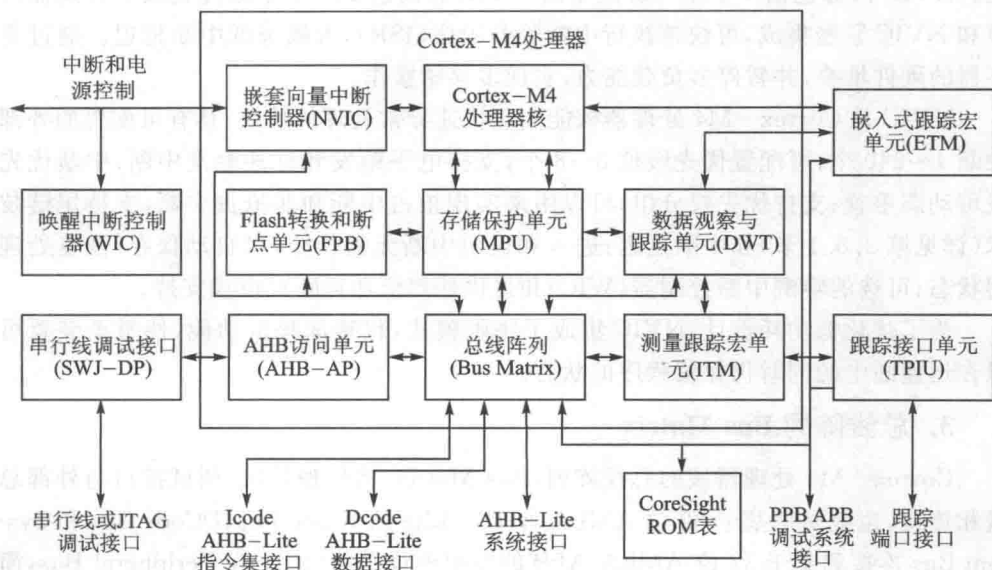


图 1 - 2 Cortex - M4 系统结构图

1. 处理器内核

Cortex - M4 是采用 ARM v7 - M 体系结构来实现的,使用 Thumb/Thumb - 2 指令集。结构上采用哈佛结构、三级流水线,可在单周期内完成 32 位乘法和采用硬件除法。具有 Thumb(正常指令状态)、Debug(调试状态)两种操作状态和 Handler(异常处理)、Thread(普通应用)两种操作模式,能够快速进入和退出中断服务程序,支持 ARM v6 类型的 BE8/LE(大端/小端数据存放形式)和 ARM v6 非对齐访问方式。



Cortex-M4 核内部寄存器包括有 13 个通用 32 位寄存器、链接寄存器 LR、程序计数器 PC、程序状态寄存器 xPSR 和 2 个堆栈指针寄存器。由于采用了哈佛操作模式, Cortex-M4 核可同时存取指令和数据, 其存储器访问接口由存取单元(Load Store Unit, 简称 LSU)和 1 个 3 字的预取单元(Prefetch Unit, 简称 PFU)组成。其中, LSU 用于分离来自 ALU 的存取操作; PFU 用于预取指令, 每次取一个字, 可以是 2 条 Thumb 指令、1 条字对齐的 Thumb-2 指令、1 条 Thumb 指令加半条半字对齐的 Thumb-2 指令、2 个半条半字对齐的 Thumb-2 指令。PFU 的预取地址必须是字对齐的, 如果 1 条 Thumb-2 指令是半字对齐的, 预取这条指令需要 2 次预取操作。不过由于 PFU 具有 3 个字的缓存, 可以确保预取第 1 条半字对齐的 Thumb-2 指令只需要 1 个延迟周期。

2. 嵌套矢量中断控制器 NVIC

在 Cortex-M4 处理器核中紧密集成了可配置的 NVIC, 提供业界领先的中断性能。NVIC 内部包括 1 个不可屏蔽中断(NMI)和高达 256 个中断优先级。处理器内核和 NVIC 紧密集成, 可快速执行中断服务程序(ISR), 大幅降低中断延迟。通过寄存器的硬件堆叠, 并暂停多负载能力, 实现多存储操作。

NVIC 是 Cortex-M4 处理器核能实现快速异常处理的关键, 具有可配置的外部中断 1~240 个; 可配置优先级位 3~8 个; 支持电平触发和脉冲触发中断; 中断优先级可动态重置; 支持优先权分组, 可以用来实现抢占中断和非抢占中断, 支持尾链技术(详见第 3.8.1 节)和中断延迟; 进入和退出中断无需指令; 可自动保存/恢复处理器状态; 可选的唤醒中断控制器(WIC)和提供外部低功耗睡眠模式支持。

为了优化低功耗设计, NVIC 集成了睡眠模式, 包括深睡眠功能, 使整个装置可以在迅速断电的同时仍保留程序的状态。

3. 总线阵列 Bus Matrix

Cortex-M4 处理器核的总线阵列(Bus Matrix)将处理器核、调试接口与外部总线相连接, 也就是把基于 32 位 AMBA AHB-Lite 的 ICode Bus、DCode Bus 和 System Bus 连接到基于 32 位 AMBA APB 的专用外设总线(Private Peripheral Bus, 简称 PPB)上。同时总线矩阵还提供非对齐数据访问方式和位段(Bit Banding)技术, 使得处理器核对片上外围设备的访问速度有了很大提高。

4. Flash 转换及断点单元 FPB

Cortex-M4 处理器核的 Flash 转换是指当 CPU 访问的某条指令匹配一个特定的 Flash 地址时, 将该地址重映射到 SRAM 中指定的位置, 从而取指后返回的是另外的值。此外, 匹配的地址还能用来触发断点事件。

FPB 有 8 个比较器, 用来产生从代码空间到系统空间转换访问(patch accesses)的硬件断点, 用于调试。其中 6 个可独立配置的指令比较器, 用于转换从代码空

间到系统空间的指令预取,或执行硬件断点。另外,2个常量比较器用于转换从代码空间到系统空间的常量访问。

5. 数据观测与跟踪单元 DWT

Cortex - M4 处理器核的 DWT 以及后面介绍的 ETM、ITM、TPIU、SW/SW - DP 单元都属于 ARM CoreSight 跟踪调试体系结构的模块,可以灵活配置使用。其中, DWT 可以设置数据观测点,参与实现调试功能。

DWT 有 4 个比较器,可配置为硬件断点、ETM 触发器、PC 采样事件触发器或数据地址采样触发器。另外,DWT 有计数器或数据匹配事件触发器用于性能剖析。DWT 还可配置用于在设定的时间间隔发出 PC 采样信息,还可发出中断事件信息。

6. 测量跟踪宏单元 ITM

Cortex - M4 处理器核的 ITM 是一个应用驱动跟踪源,支持应用事件跟踪和 printf 类型的调试。它支持如下跟踪信息源:

① 软件跟踪。软件可直接写 ITM 单元内部的激励寄存器,使之向外发送相关信息包。

② 硬件跟踪。DWT 产生信息包,由 ITM 向外发送。

③ 时间戳。ITM 可产生与所发送信息包相关的时间戳包,并对外发送。

④ 全局系统时间戳。ITM 可以产生一个全系统的 48 位计数值用作时间戳,并对外发送。

7. 串行线调试接口 SW / SWJ - DP

Cortex - M4 处理器的调试接口 SW/SWJ - DP 可以提供对处理器内所有寄存器和存储器的访问。该调试接口通过处理器内部的 AHB - AP (Advanced High - performance Bus Access Port)来实现调试访问。对于此调试接口而言,外部调试口有两种可能的实现方法。

一种是串行 JTAG 调试接口 SWJ - DP (Serial Wire JTAG Debug Port), SWJ - DP 是 JTAG - DP 和 SW - DP (Serial Wire Debug Port)的结合;另一种是 SW - DP 调试口,该调试口通过 2 个引脚(clock、data)实现与处理器内部 AHB - AP 的接口。

8. 嵌入式跟踪宏单元 ETM

Cortex - M4 处理器核的 ETM 单元是一个仅支持指令跟踪的低成本高速跟踪宏单元,对于 Cortex - M4 而言是可选的。通过 ETM 发出的数据,可以重构程序执行过程。不过 ETM 的数据量非常大,对于外部硬件跟踪设备和工具软件的要求都比较高。

9. 跟踪接口单元 TPIU

Cortex - M4 处理器核的 TPIU 单元是 ITM 单元、ETM 单元与片外跟踪分析器



之间传递跟踪数据的桥梁。该 TPIU 单元兼容 CoreSight 调试体系结构,如果还需要添加额外功能,可用 CoreSight TPIU 替代。TPIU 可配置为仅支持 ITM 调试跟踪,由于 ITM 数据量不大,因此可采用低成本的串行跟踪形式。也可配置为支持 ITM 和 ETM 的跟踪调试,这时需使用高带宽的跟踪接口及设备。

10. 存储器保护单元 MPU

MPU 是 Cortex - M4 处理器核中一个可选的模块,通过定义和检查存储区域的属性来实现存储保护,以改善嵌入式系统的可靠性实现安全操作。带有此单元的 Cortex - M4 处理器核,支持标准 ARM v7 保护存储系统结构模型。MPU 可以提供以下支持:

- ① 存储保护,包含 8 个存储区域和 1 个可选的后台区域。
- ② 保护区域重叠。
- ③ 访问允许控制。
- ④ 向系统传递存储器属性。

通过以上支持,MPU 可以实现存储管理优先规则、分离存储过程和实现存储访问规则。

1.3 Cortex - M4 寄存器组织

由于 Cortex - M4 处理器核采用 ARM v7 - M 架构,与之前的 ARM v4、ARM v5、ARM v6 等体系结构有较大的不同。本节将主要介绍 Cortex - M4 处理器核的寄存器组织。

1. 通用寄存器

Cortex - M4 处理器核内部具有 16 个通用寄存器。其中 r0~r12 没有特定的功能,绝大多数指令都可以使用。它们被分为两组,其中寄存器 r0~r7 为低寄存器,可被所有指令访问;寄存器 r8~r12 为高寄存器,可以被所有的 32 位指令访问,但不能被 16 位指令访问。另外 r13、r14 和 r15 寄存器分别有以下特定功能。

寄存器 r13 被用于栈指针(SP),由于 SP 忽略了位[1:0],因此它自动对齐为 1 个字,即 4 字节。该寄存器是分组的,分别为 Main Stack Pointer(MSP)和 Process Static Pointer(PSP)。Handler 模式一般使用 MSP,在 Thread 模式下可以配置选择使用 MSP 或 PSP:

- 0 = Main Static Pointer;
- 1 = Process Static Pointer。

寄存器 r14 是子程序链接寄存器(LR),当执行带链接的跳转指令 BL 或带链接及状态切换的跳转指令 BLX 时,LR 寄存器将保存 PC 作为返回地址,LR 也用于异常返回。其他时候,可以把 r14 寄存器当作通用寄存器来使用。

寄存器 r15 为程序计数器,其位[0]总为 0,因为指令是按字或半字对齐。

2. 状态寄存器

在系统层,处理器状态可分为应用(Application)、中断(Interrupt)和执行(Execution)三种类型,与之对应有 3 个程序状态寄存器,即 APSR、IPSR 和 EPSR。这 3 个寄存器实际上是合成一个的,可以使用 MRS 和 MSR 指令来访问。在进入异常时,处理器会把该寄存器的信息入栈。

(1) APSR 寄存器

Cortex - M4 处理器核的 APSR 寄存器包含程序的条件标志,如图 1-3 所示。该寄存器中各位的含义见表 1-1。

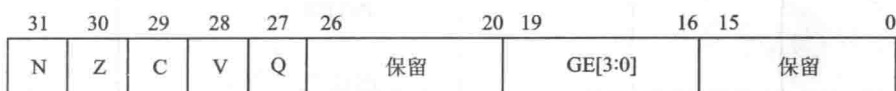


图 1-3 APSR 寄存器的条件标志位

表 1-1 APSR 寄存器各位域定义

位区域	名称	说明
[31]	N	负数或小于标志: 1=负数或小于;0=正数或大于
[30]	Z	零标志: 1=结果为零;0=结果非零
[29]	C	进位/借位标志: 1=进位或借位;0=无进位或借位
[28]	V	溢出标志: 1=溢出;0=无溢出
[27]	Q	粘性饱和标志
[26:20]	—	保留
[19:16]	GE[3:0]	每个字节对应的大于或等于标志
[15:0]	—	保留

(2) IPSR 寄存器

Cortex - M4 处理器核的 IPSR 寄存器包含当前正在执行的中断服务子程序(ISR)号,如图 1-4 所示。该寄存器中各位的含义见表 1-2。



图 1-4 IPSR 寄存器的条件标志位