■ 周润景 编著

ARM7

嵌入式系统设计与仿真

基于Proteus、Keil与IAR

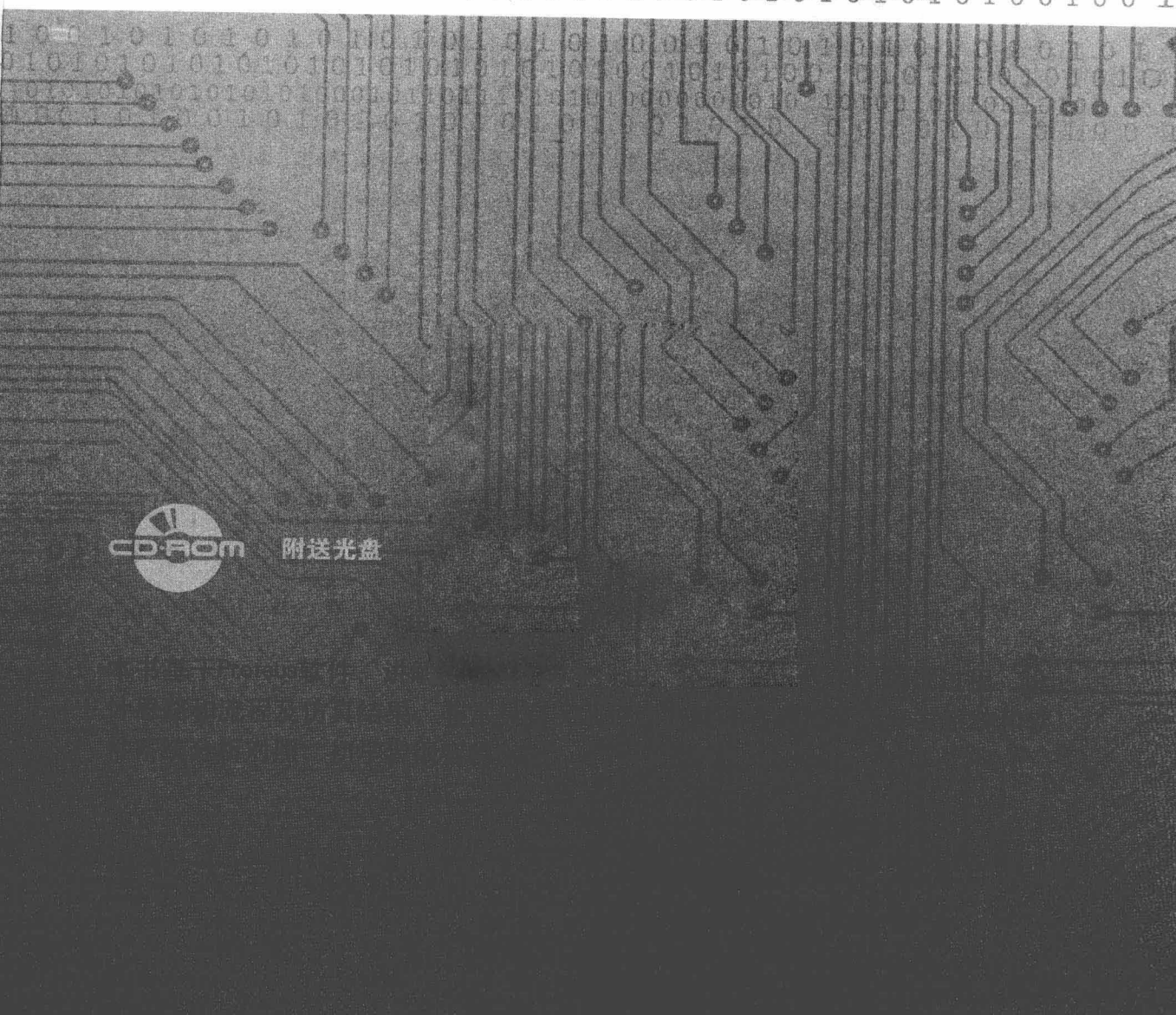
附送光盘

清华大学出版社

周润景 编著

ARM7 嵌入式系统设计与仿真

基于Proteus、Keil与IAR



CD-ROM 附送光盘

内 容 简 介

本书结合动态仿真工具软件 Proteus 和编译软件 IAR，以读者最容易理解的方式介绍了如何使用软件平台设计 ARM7 嵌入式系统。ARM 芯片选用了 Philips 公司的 LPC2138，通过实例使读者掌握嵌入式系统的设计方法。全书分为 6 章，包括嵌入式系统概述、ARM 体系结构、LPC2138 硬件结构、Proteus 7.8 软件入门设计、Keil for ARM 程序设计与电路仿真、IAR for ARM 程序设计与电路仿真，每章中都有大量的实例和相关习题，方便读者学习。

本书可作为从事嵌入式系统设计的学生、教师、科研人员以及广大电子爱好者的参考资料，对日常教学、学生实验、课程设计、毕业设计以及电子竞赛等都有很大帮助。

图书在版编目（CIP）数据

ARM 7 嵌入式系统设计与仿真——基于 Proteus、Keil 与 IAR / 周润景编著. —北京：清华大学出版社，2012.1
ISBN 978-7-302-27741-5

Ⅰ.①A…　Ⅱ.①周…　Ⅲ.①微处理器，ARM7－系统设计②微处理器，ARM7－系统仿真　Ⅳ.①TP332

中国版本图书馆 CIP 数据核字（2011）第 280177 号

责任编辑：袁金敏　薛　阳
责任校对：徐俊伟
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：20.25　**字　数**：509 千字
附光盘 1 张
版　　次：2012 年 1 月第 1 版　　**印　　次**：2012 年 1 月第1次印刷
印　　数：1～4000
定　　价：49.00 元

产品编号：045148-01

前　言

当代生活的每一个角落都有嵌入式设备的存在，如数码相机、移动电话、TV 机顶盒和掌上电脑等，这些设备多采用 32 位 RISC 嵌入式处理器作为核心部件，其中基于 ARM 核的嵌入式处理器独占鳌头，在 32 位 RISC 处理器中占据超过 75%的市场份额。

在传统的嵌入式系统学习中，嵌入式开发平台是必不可少的，而资源少的开发平台价格便宜，但功能较少；资源多的开发平台，价格又不菲。英国 Labcenter 公司推出了适合嵌入式设计仿真与开发平台的软件 Proteus，在该软件中可以根据需要搭建开发平台，将编译好的目标代码加载到芯片中。使用 Proteus 软件可以完全脱离硬件平台来学习嵌入式系统，可以说是嵌入式系统学习的一次革命。

本书以 Philips 公司的 LPC2138 微控制器为例，结合 Keil for ARM 和 IAR 开发工具，以大量实例介绍如何在 Proteus 中搭建硬件开发平台进行源代码级调试。全书共分 6 章，在内容安排上依照循序渐进的原则。

第 1 章是概述篇。读者可以初步了解嵌入式系统的概况，从不同的角度认识 ARM 嵌入式处理器，包括嵌入式微处理器、嵌入式微控制器、嵌入式 DSP 处理器和嵌入式片上系统等。

第 2 章重点介绍 ARM 体系结构，从不同方面介绍 ARM 微处理器体系结构，详细阐述了存储器、处理器、内部寄存器和程序状态寄存器的内容，对异常、中断延迟、复位、存储器映射及存储器映射 I/O 做了深入讲解。以直观的方法介绍了寻址方式、ARM7 指令集等，并针对不同的接口——协处理器接口、调试接口、ETM 接口进行了说明。

第 3 章讲解了 LPC2138 的硬件结构，阐述了系统控制模块、存储器加速模块的相关功能，介绍了微处理器引脚配置以及引脚连接典型模块——GPIO、UART、I^2C 接口、SPI、定时器、脉宽调制 PWM、A/D 转换器、实时时钟 RTC 和看门狗定时器（WDT）。以方便、直观、形象的方式进行了清晰而详尽的讲解。

第 4 章集中介绍了动态仿真软件 Proteus，通过一个简单的案例阐述了该软件的使用方法，突出了该软件对复杂电路仿真的应用，Proteus 基于界面友好且功能齐全的 Windows 操作平台，为用户提供了一个嵌入式微处理器 LPC2138 的设计和开发环境。

第 5 章以一个实际的例子详细阐述了 Keil for ARM 软件的使用方法，针对 LPC2138 不同模块的相关功能，运用大量案例直观说明 ARM 控制系统的设计和仿真，并在每个案例后结合 Proteus 仿真软件进行仿真，将理论和实际紧密结合在一起，更加深入理解设计的应用。

第 6 章着重讲解了 IAR Embedded Workbench for ARM version 软件的使用方法，同第 4 章一样通过一个实际案例讲解了软件的使用方法，包括相关参数的配置及程序的下载，并结合 LPC2138 不同模块的相关功能，运用大量实例介绍了 LPC2138 控制系统的设计和仿真，并在每个案例后结合 Proteus 仿真软件进行仿真。

在编写过程中参考了许多书籍、文章和标准等，这些参考文献使作者深受启发，在此向各位作者表示感谢。

本书共分 6 章，刘梦男编写了第 1 章，其余由周润景编写，全书由周润景定稿。

由于作者水平有限，书中错误与不妥之处在所难免，敬请广大读者批评指正。

作者

2011.8

目　录

第 1 章　嵌入式系统概述

1.1　嵌入式系统简介

根据 IEEE（国际电气和电子工程师协会）的定义，嵌入式系统是“控制、监视或者辅助设备、机器和车间运行的装置”（全称为 devices used to control，monitor，or assist the operation of equipment，machinery）。这主要是从应用上加以定义的，由此可以看出嵌入式系统是软件和硬件的综合体，可以涵盖机械等附属装置。

目前国内一个普遍的定义：以应用为中心、以计算机技术为基础，软硬件可裁减，适于应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。还有另一种定义：嵌入式系统是设计完成复杂功能的硬件和软件，并使其紧密耦合在一起的计算机系统。术语“嵌入式”反映了这些系统通常是更大系统中的一个完整部分，称为嵌入的系统。两种定义的出发角度不同，前者是从技术角度来定义的，后者是从系统角度来定义的。由于嵌入式系统本身是一个外延很广的名词，凡是与产品结合在一起的具有嵌入式特点的控制系统都可以叫嵌入式系统，很难给它下一个准确的定义。因此，目前通常把嵌入式系统概念的重心放在“系统”（即操作系统）上，指能够运行操作系统的软硬件综合体。总体上嵌入式系统可以划分成硬件和软件两部分，硬件一般由高性能的微处理器和外围接口电路组成，软件一般由实时操作系统和其上运行的应用软件构成，软件和硬件之间由所谓的中间层（BSP 层，板级支持包）连接。

一般而言，嵌入式系统的架构可以分成 4 个部分：处理器、存储器、输入/输出（I/O）和软件，如图 1.1 所示。

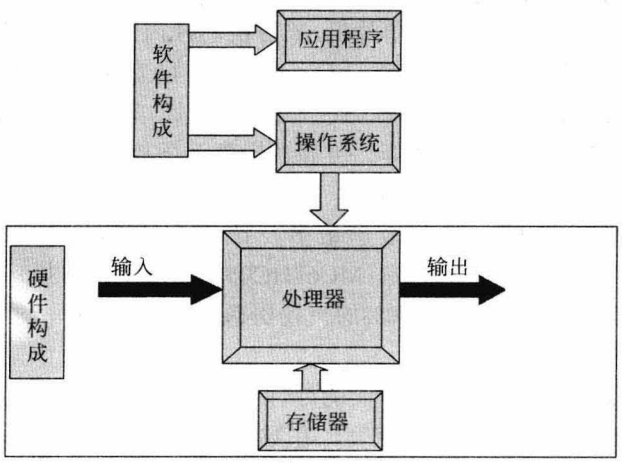


图 1.1　嵌入式系统的架构

1.2 嵌入式处理器

1.2.1 嵌入式处理器简介

从硬件方面讲，嵌入式系统的核心是嵌入式处理器，据不完全统计，全世界嵌入式处理器的品种数量已经超过 1000 多种，流行体系结构有 30 多个，其中 8051 体系占大多数。生产 8051 单片机的半导体厂家有 20 多个，共 350 多种衍生产品，仅 Philips 就有近 100 种。近年来，嵌入式微处理器的主要发展方向是小体积、高性能、低功耗，专业分工也越来越明显，出现了专业的 IP（Intellectual Property Core，知识产权核）供应商，如 ARM、MIPS 等，它们提供优质、高性能的嵌入式微处理器内核，由各半导体厂商生产面向各个应用领域的芯片。

嵌入式微处理器有许多种流行的处理器核，芯片制造商一般都基于这些处理器核生产不同型号的芯片。一般可以将嵌入式处理器分成 4 类，即嵌入式微处理器（MicroProcessor Unit，MPU）、嵌入式微控制器（MocroController Unit，MCU）、嵌入式 DSP 处理器（Digital Signal Processor，DSP）和嵌入式片上系统（System on Chip，SoC）。

❑ 嵌入式微处理器

嵌入式微处理器的基础是通用计算机中的 CPU，它的特征是 32 位以上的处理器，具有较高的性能，当然其价格也相当高。为了满足嵌入式应用的特殊要求，虽然嵌入式微处理器在功能上和标准微处理器基本上是一样的，但一般在工作温度、抗电磁干扰及可靠性等方面都做了各种增强。

与工业控制计算机相比，嵌入式微处理器具有体积小、重量轻、成本低、可靠性高等优点。嵌入式微处理器目前主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS 和 ARM/StrongARM 系列等。

❑ 嵌入式微控制器

嵌入式微控制器最典型的代表是单片机，单片机芯片内部集成了 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制输出、A/D、D/A、Flash、EEPROM 等各种必要功能和外设。与嵌入式微处理器相比，微控制器的最大特点是单片机，体积大大减小，从而使功耗和成本下降、可靠性提高。微控制器是目前嵌入式系统工业的主流。微控制器的片上外设资源一般比较丰富，适合于控制，因此称为微控制器。

嵌入式微控制器目前的品种和数量最多，比较有代表性的包括 8051、P51XA、MCS-251、MCS96/196/296、C166/167、MC68HC05/11/12/16、68300 和数目众多的 ARM 芯片。目前，MCU 占嵌入式系统约 70%的市场份额。

❑ 嵌入式 DSP 处理器

DSP 处理器是专门用于信号处理方面的处理器，其在系统结构和指令上进行了特殊设计，具有很高的编译效率和指令执行速度。在数字滤波、FFT、频谱分析等仪器上，DSP 获得了大规模的应用。

嵌入式 DSP 处理器比较有代表性的产品是 TI 的 TMS320 系列和 Motorola 的 DSP56000

系列。

❑ 嵌入式片上系统

片上系统（SoC）最大的特点是成功实现了软硬件无缝结合，直接在处理器片内嵌入操作系统的代码模块。各种通用处理器内核将作为 SoC 设计公司的标准库，和许多其他嵌入式系统外设一样，成为 VLSI 设计中一种标准的器件，用标准的 VHDL 等语言描述，存储在器件库中。用户只需定义出其整个应用系统，仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的器件以外，整个嵌入式系统大部分均可集成到一块或几块芯片中去，应用系统电路板将变得很简洁，对于减小体积和功耗、提高可靠性非常有利。

SoC 可以分为通用和专用两类。通用系列包括 Infineon 的 TriCore、Motorola 的 M-Core、某些 ARM 系列器件、Echelon 和 Motorola 联合研制的 Neuron 芯片等。专用 SoC 一般专用于某个或某些系统中，不为一般用户所知。一个有代表性的产品是 Philips 的 Smart XA，它将 XA 单片机内核和支持超过 2048 位复杂 RSA 算法的 CCU 单元制作在一块硅片上，形成一个可加载 Java 或 C 语言的专用 SoC，可用于公众互联网（如 Internet）安全方面。

1.2.2　ARM 处理器简介

ARM（Advanced RISC Machines）是全球领先的 16/32 位 RISC 微处理器的知识产权设计供应商，ARM 公司通过转让高性能、低成本、低功耗的 RISC 微处理器、外围和系统芯片设计技术给合作伙伴，使他们能用这些技术来生产各具特色的芯片。ARM 处理器小体积、低功耗、低成本。

目前，应用比较多的是 ARM7 系列、ARM9 系列、ARM9E 系列、ARM10 系列、ARM11 系列、SecurCore 系列和 Intel 的 StrongARM、XScale 系列。

ARM7TDMI 基于 ARM 体系结构 V4 版本，是目前低端的 ARM 核，具有广泛的应用，其最显著的应用为数字移动电话。ARM7TDMI 使用流水线以提高处理器指令的流动速度。流水线允许几个操作同时进行，以及处理和存储系统连续操作。ARM7TDMI 使用三级流水线，因此，指令的执行分成三个阶段——取指、译码和执行。ARM7TDMI 核是冯·诺依曼体系结构，使用单一 32 位数据总线传送指令和数据，只有加载、存储和交换指令可以访问存储器中的数据。

ARM7TDMI-S 是 ARM7TDMI 的可综合版本（软核）。对应用工程师来说，除非芯片生产商对 ARM7TDMI-S 进行了裁减，否则在逻辑上 ARM7TDMI-S 与 ARM7TDMI 没有太大区别，其编程模型与 ARM7TDMI 一致。

第 2 章　ARM 体系结构

2.1　ARM 处理器结构

2.1.1　ARM 处理器结构概述

ARM 是一种精简指令集计算机，它具有外形非常小但性能高的结构，它的数据处理操作只针对寄存器的内容，而不直接对存储器进行操作。由寄存器内容和指令域决定简单的寻址模式，每一条数据处理指令都对算术逻辑单元（ALU）和移位器控制，以实现对 ALU 和移位器的最大利用，地址自动增加和自动减少的寻址模式实现了程序循环的优化。所有指令的条件执行实现最快速的代码执行，使 ARM 处理器在高性能、低代码规模、低功耗和小的硅片尺寸方面取得良好的平衡。

ARM 处理器具有优异的性能且功耗很低，使用很少数量的门，基于精简指令集计算机原理设计，指令集和相关的译码机制比复杂指令集计算机要简单得多。

2.1.2　流水线结构

ARM 处理器通过使用流水线结构来增加处理器指令流的速度，这样可使几个操作同时进行，并使处理和存储器系统连续操作，提供 0.9 MIPS/MHz 的指令执行速度。

流水线使用 3 个阶段执行：

（1）取指；

（2）译码；

（3）执行。

程序计数器（PC）指向被取指的指令，在执行一条指令的同时，对下一条指令进行译码，并将第 3 条指令从存储器中取出。

2.2　存储器

ARM 处理器的指令和数据共用一条 32 位总线。只有装载、存储和交换指令可以对存储器中的数据进行访问。数据可以是 8 位字节、16 位半字或者 32 位字。字必须分配为占用 4 字节，而半字必须分配为占用 2 字节。ARM 处理器的存储器接口设计可以使潜在的性能得到实现，这样减少了存储器使用。对速度有严格要求的控制信号使用流水线，这些控制信号使许多片内和片外存储器技术所支持的“快速突发访问模式”得到利用。

ARM 处理器的存储器周期有 4 种基本类型：

（1）内部周期；

（2）非连续的周期；

（3）连续的周期；

（4）协处理器寄存器传输周期。

2.3　处理器

ARM 处理器支持 8 位字节、16 位半字、32 位字的数据类型。其内核结构包含 32 位 ARM 指令集和 16 位 Thumb 指令集，有两种操作状态：ARM 状态和 Thumb 状态，在 Thumb 状态中，程序计数器使用 bitl 来选择切换半字。使用 BX 指令内核的操作状态在 ARM 状态和 Thumb 状态之间进行行切换，

LPC2138 结构支持 7 种处理器模式：用户模式、快中断模式、中断模式、管理模式、中止模式、未定义模式和系统模式。除用户模式外，其他模式均为特权模式。ARM 内部寄存器和一些片内外设在硬件设计上只允许特权模式下访问。特权模式可自由地切换处理器模式，而用户模式不能直接切换到别的模式。具体处理器模式说明如表 2.1 所示。

表 2.1　处理器模式

处理器模式	说明	备注
用户（usr）	正常程序工作模式	不能直接切换到其他模式
快中断（fiq）	支持高速数据传输及通道处理	FIR 异常响应时，进入此模式
中断（irq）	用于通用中断处理	IRQ 异常响应时，进入此模式
管理（svc）	操作系统保护代码	系统复位和软件中断响应时，进入此模式
中止（abt）	用于支持虚拟内存和存储器保护	在 ARM7TDMI 中没有大用处
未定义（und）	支持硬件协处理器的软件仿真	未定义指令异常响应时，进入此模式
系统（sys）	用于支持操作系统的特权任务等	与用户类似，但具有可以直接切换到其他模式等特权

有 5 种处理器模式称为异常模式，它们是：

（1）快中断模式；

（2）中断模式；

（3）管理模式；

（4）中止模式；

（5）未定义模式。

以上模式除了可通过程序切换进入外，也可由特定的异常进入。当特定的异常出现时，处理器进入相应的模式。每种模式都有某些附加的寄存器，以避免异常退出时，用户模式的状态不可靠。

系统模式与用户模式一样不能由异常进入，使用与用户模式完全相同的寄存器。由于它是特权模式，因而不受用户模式的限制。运用这个模式，操作系统访问用户模式的寄存

器比较方便。另外，操作系统的一些特权任务也可使用这个模式，以访问一些受控资源，而需要考虑异常出现时任务状态变得不可靠。

2.4 内部寄存器

ARM 处理器内部有 37 个用户可见的寄存器：

（1）31 个通用 32 位寄存器，在 ARM 公司文件中它们的名称分别为 R0～R15、R13_svc、R14_svc、R13_abt、R14_abt、R13_und、R14_und、R13_irq、R14_irq 和 R8_fiq～R14_fiq。

（2）6 个状态寄存器，在 ARM 公司文件中，它们的名称分别为 CPSR、SPSR_svc、SPSR_abt、SPSR_und、SPSR_irq 和 SPSR_fiq。

这些寄存器并不是全都可以在同一时间被访问，处理器状态和操作模式决定了访问哪些寄存器。

2.4.1 各模式可访问寄存器

在 ARM 状态中，16 个通用寄存器和 1 个或 2 个状态寄存器可在任何时候同时被访问。在特权模式中，与模式相关的分组寄存器可以被访问。表 2.2 所示为所能访问的寄存器。

表 2.2 ARM 状态各模式下的寄存器

寄存器类别	寄存器在汇编中的名称	各模式实际访问的寄存器						
		用户	系统	管理	中止	未定义	中断	快中断
通用寄存器和程序计数器	R0(al)	R0						
	Rl(a2)	Rl						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(vl)	R4						
	R5(v2)	R5						
	R6(v3)	R6						
	R7(v4)	R7						
	48(v5)	R8						R8_fiq
	R9(SB，v6)	R9						R9_fiq
	R10(SL，v7)	Rl0						Rl0_fiq
	R11(FP，v8)	Rll						Rll_fiq
	R12(IP)	R12						R12_fiq
	R13(SP)	R13	R13_svc		R13_abt	R13_und	R13_irq	R13_fiq
	R14(LR)	R14	R14_svc		R14_abt	R14_und	R14_irq	R14_fiq
	R15(PC)	R15						
状态寄存器	CPSR	CPSR						
	SPSR	无	SPSR_svc		SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

2.4.2　通用寄存器

1．数据或地址保存寄存器

寄存器 R0～R13 为保存数据或地址的通用寄存器，其中寄存器 R0～R7 为未分组的寄存器。这意味着对于任何处理器模式，它们中的每一个都对应于相同的 32 位物理寄存器。它们是完全通用的寄存器，不会被体系结构作为特殊的用途，并且可用于任何使用通用寄存器的指令。

2．分组寄存器

寄存器 R8～R14 为分组寄存器。它们所对应的物理寄存器取决于当前的处理器模式。几乎所有允许使用通用寄存器的指令都允许使用分组寄存器。寄存器 R8～R12 有两个分组的物理寄存器。一个用于除 FIQ 模式之外的所有寄存器模式（R8～R12），另一个用于 FIQ 模式（R8_fiq～R12_fiq）。寄存器 R8～R12 在 ARM 体系结构中没有特定的用途。FIQ 所单独使用的这些寄存器可实现快速的中断处理。寄存器 R13 和 R14 分别有 6 个分组的物理寄存器。一个用于用户和系统模式，其余 5 个分别用于 5 种异常模式。

3．堆栈指针 R13

寄存器 R13 通常作为堆栈指针 SP。在 ARM 指令集中，没有以特殊方式使用 R13 的指令或其他功能，而在 Thumb 指令集中存在使用 R13 指令的情况。

每个异常模式都有其自身的 R13 分组版本，它通常指向由异常模式所专用的堆栈。在入口处，异常处理程序通常将其他要使用的寄存器值保存到这个堆栈。通过返回时将这些值重装到寄存器中，异常处理程序可确保异常发生时的程序状态不会被破坏。

4．链接寄存器 R14

寄存器 R14（也称为链接寄存器或 LR）在结构上有两个特殊功能：

在每种模式下，模式自身的 R14 版本用于保存子程序返回地址。当使用 BL 或 BLX 指令调用子程序时，R14 设置为子程序返回地址。子程序返回通过将 R14 复制到程序计数器来实现。

通常通过执行下列指令的两种方式：

```
MOV  PC,LR
BX   LR
```

在子程序入口，使用下列形式的指令将 R14 存入堆栈：

```
STMFD SP ! , { <registers> ,LR}
```

并使用匹配的指令返回：

```
LDMFD  SP ! , { <registers>,PC}
```