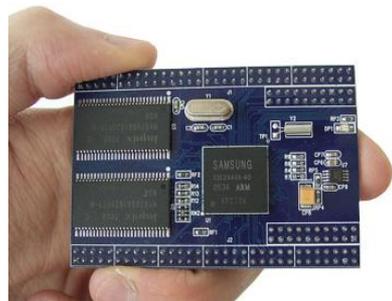


# 嵌入式系统原理及接口技术-2

## 2 ARM9体系结构

- 嵌入式系统硬件平台的核心部件是微处理器。
- 在设计嵌入式系统时，可采用的微处理器芯片种类有很多。
- ARM9是ARM公司推出的、并被广泛使用的嵌入式微处理器核。
- 国家上有许多著名的电子公司生产基于ARM9核的CPU芯片，如：三星电子（Samsung）、Atmel、英特尔、飞思卡尔（Freescale）、恩智浦（NXP）、德州仪器（TI）等。



# ARM微处理器系列介绍

- ARM微处理器核的体系结构从出现开始到现在，经过了多次较大的改进，并还在不断地完善和发展中，且其指令集也在不断地修改和增加。
- 目前，有2大类微处理器核的命名。一是以“ARM”字符为前缀的微处理器系列，另一是以“Cortex”字符为前缀的微处理器系列。



- 以字符“ARM”为前缀命名的ARM微处理器核有许多系列，从ARM1系列一致到ARM11系列。
- ARM系列中的一些符号含义：
  - T: 支持高密度16位的压缩Thumb指令集；
  - D: 支持片上调试；
  - M: 支持64位除法；
  - I: 具有Embedded-ICE（即嵌入式的ICE），支持片上断点；
  - E: 增强性，具有Java、DSP等增强的处理功能；
  - S: 表示是综合处理器；
  - J: 表示具有Jazelle技术，该技术支持。

如：ARM7TDMI、ARM920T

- Cortex系列是ARM公司为其微处理器核进行命名的另外一个系列，自采用ARMv7指令集架构的微处理器核开始，其名称前缀采用“Cortex”，而不再采用“ARM”。换句话说，ARM11是以“ARM”为前缀的最后一个微处理器核系列的名称，以后不再沿用这种命名方式，而是以“Cortex”来进行命名。
- Cortex系列微处理器核有3大系列类型，分别是Cortex-A、Cortex-R、Cortex-M。

- ARM指令集版本号含义：
- ARM公司给这些在不同阶段体系结构中使用的指令集定义了版本号，版本号的名称为：ARMvx（符号x代表数字，即1、2、3、...），如ARMv4、ARMv7等即是ARM的指令集版本号。
- ARMv4版本是目前使用非常广泛的一个ARM指令集，ARM9等系列的微处理器芯片均采用了该版本的架构。ARM10、Xscale等系列的微处理器芯片采用了ARMv5版本指令集。ARM11系列的微处理器芯片采用了ARMv6版本指令集。Cortex系列的微处理器芯片采用了最新的ARMv7版本指令集。

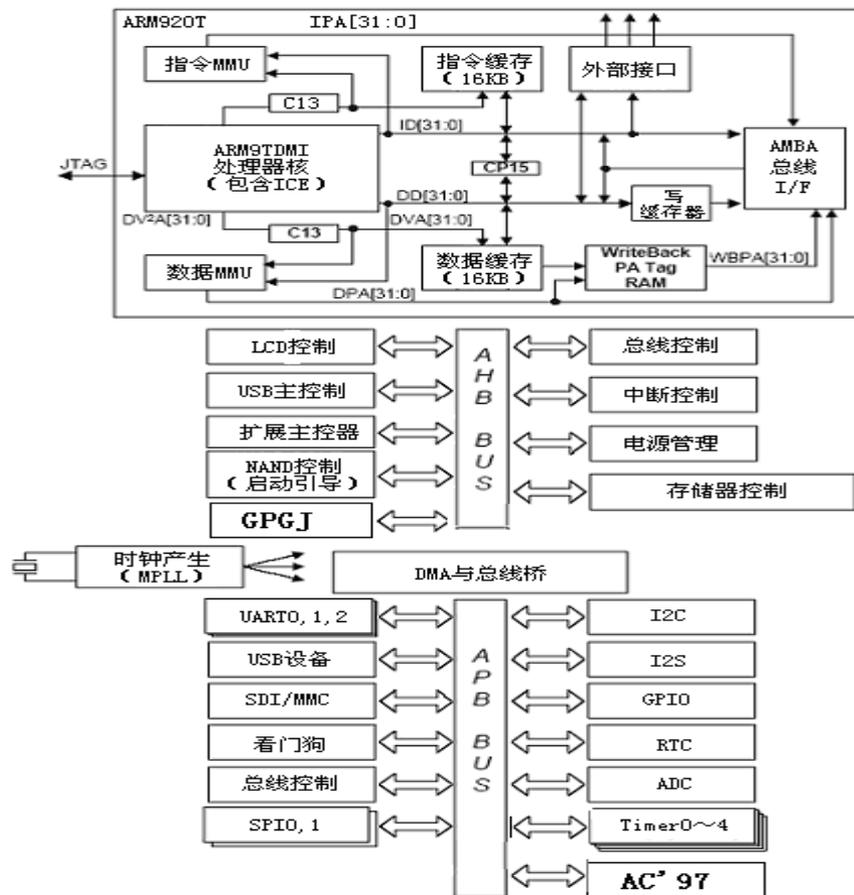
# ARM9微处理器特征

- ARM公司微处理器核的系列及种类非常多，不同系列之间，微处理器核的体系结构特征有所不同，所基于的ARM指令架构（即ARM的版本号）也不同。
- ARM9系列微处理器是目前在我国用得非常广泛的、拥有高性价比的一种嵌入式微处理器，非常适合应用在要求低成本、低功耗的消费类产品中，例如：手机、PDA、MP4播放器，也被大量地应用在工业控制、智能仪器仪表、医疗仪器等领域。

- 基于ARM9处理器核的微处理器芯片：
  - 1 三星公司的S3C2440
  - 2 Atmel公司的AT91RM9200
  - 3 恩智浦公司的LPC2900
  - 4 Intel公司的PXA255
  - 5 Motorola公司的MC9328
  - 6 Cirrus Logic公司的EP93XX系列、等等。

本教材结合三星公司的S3C2440系列微处理器芯片（其内部采用了ARM920T微处理器核）来介绍ARM9微处理器的基本结构特征。

- S3C2440芯片内部逻辑结构:



# 指令流水特征

- ARM920T微处理器的指令流水采用了典型的RISC五级流水线结构，将指令的处理过程分为取指、译码、执行、访存和写回5个步骤。
- 五级流水线结构采用分离指令缓存和数据缓存的结构，即哈佛结构，将每一条指令处理分配到3个或者5个时钟周期内完成。



- 由于ARMv4架构的汇编指令集中，不同功能的指令，需要的时钟周期不一样。例如，对于寄存器操作的指令（即寄存器间数据传送指令、加减运算的指令等），所需时钟周期数为3个，而对于存储器的加载/存储指令，所需时钟周期数为5个，也就是需要增加2个时钟周期来完成访存、写回步骤。
- 在访存、写回步骤时，操作的是数据缓存，不占用指令缓存的信号总线，因此，不会阻塞指令流水。

# 指令集特征

- ARM920T的指令集是基于精简指令集计算机（RISC）原理的；
- 传统的微处理器体系结构中，指令代码的宽度（位数）和数据的宽度（位数）通常是相同的，而ARM920T的指令系统中有一种**16位**的指令集（Thumb指令集）。
- 而Thumb指令集在**32位**体系结构上实现了**16位**指令集，以提供比**16位**体系结构更高的性能和比**32位**体系结构更高的代码密度。

# ARM9的工作模式特征

- ARM9微处理器核共支持7种工作模式：

处理器模式	模式符号	意义
用户模式	Usr	正常执行程序时的处理器模式
系统模式	Sys	运行特权操作系统任务时的模式
管理模式	Svc	操作系统的保护模式
中止模式	Abt	指令或数据预取操作中中止时的模式，该模式下实现虚拟存储器或存储器保护
未定义模式	Und	当执行未定义的指令时进入该模式
IRQ模式	Irq	响应普通中断时的处理模式
FIQ模式	Fiq	响应快速中断时的处理模式

- ARM9微处理器核的工作模式是可以通过软件控制来改变的，而且，外部中断信号或系统异常处理也可以使工作模式发生改变。
- 当工作在用户模式下，ARM9微处理器核所执行的程序是不能访问那些被保护的系统资源，也不能改变微处理器核的工作模式。
- 当系统发生异常时，ARM9微处理器核的工作模式就会改变。
- 对于系统模式来说，其通用寄存器组与用户模式下的通用寄存器组是完全相同的，但它是一种特权模式，供需要访问系统资源的操作系统任务使用。

# ARM9的存储组织结构

- ARM920T核允许32位长的地址，它把存储器看成是从0x00000000地址开始的字节单元的线性组合，即一个地址对应于一个存储字节，其整个地址范围是2<sup>32</sup>个字节（8位二进制数），即4GB。
- ARM9中的地址特征：
  - 1、ARM9微处理器核的每个地址是对应于一个存储字节而不是一个存储字，但ARM9微处理器核可以访问存储字，访问存储字时，其地址应该是字对准的，既字地址可以被4整除。
  - 2、程序中若遇到分枝指令，目的地址的计算公式如下：  
目的地址 = PC值+偏移量 = 当前执行的指令地址+8+偏移量
  - 3、ARM9存储器系统的存储单元与地址的对应方式有两种：大端存储系统、小端存储系统。

# 大端存储和小端存储示意图

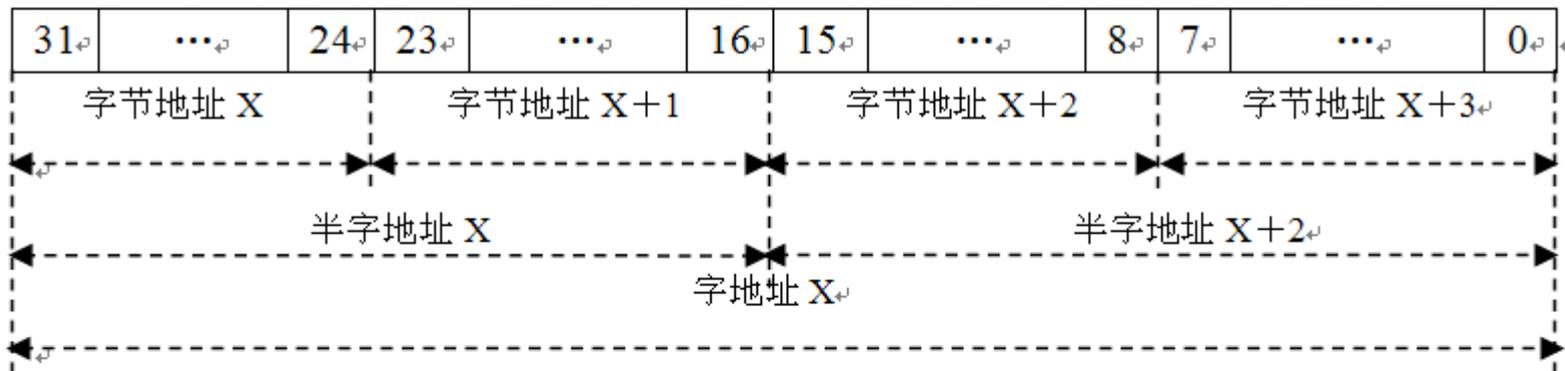


图 2-4 大端存储系统

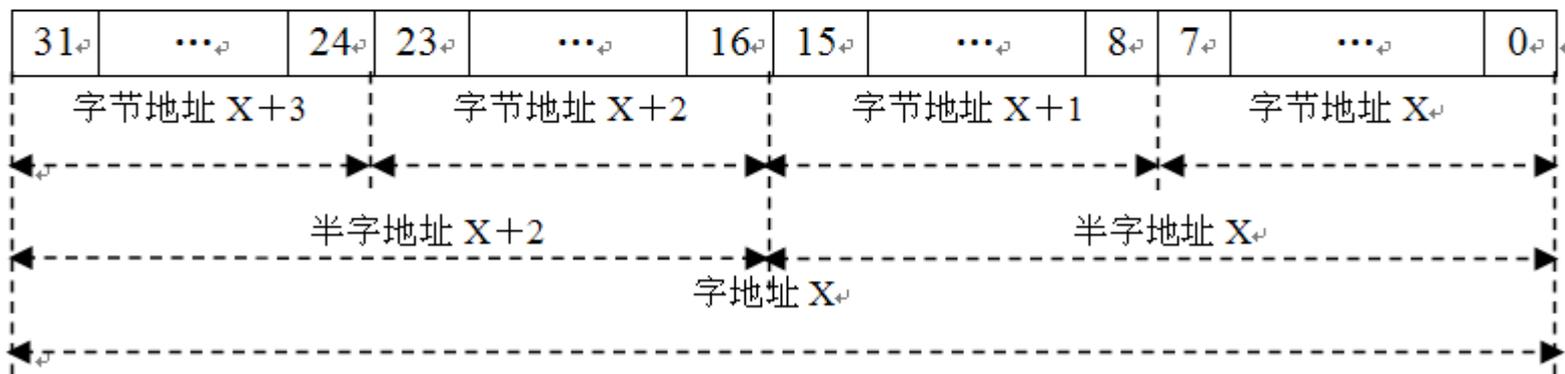


图 2-5 小端存储系统

- 大端存储系统中，字的地址对应的是该字中最高有效字节所对应的地址；
- 小端存储系统中，字的地址对应的是该字中最低有效字节所对应的地址。

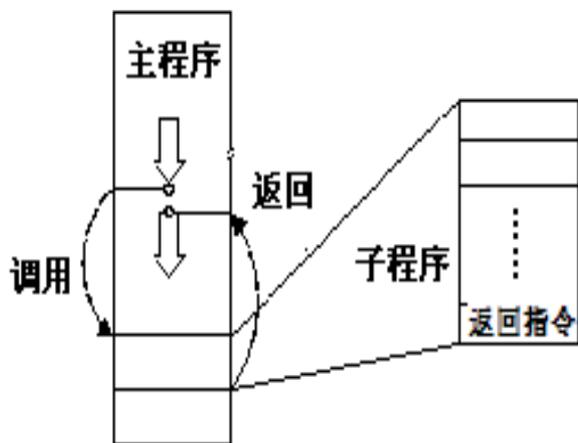
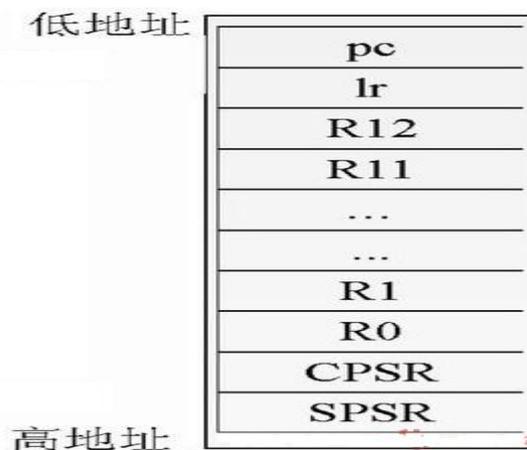
- 内部寄存器：ARM9微处理器的内部总共有37个32位的寄存器，其中31个用作通用寄存器，6个用作状态寄存器，每个状态寄存器只使用了其中的12位。这37个寄存器根据处理器的状态及其工作模式的不同而被安排成不同的组。
- 寄存器的名称是R0~R15、CPSR、SPSR
- 其中：R0~R7是不分组的寄存器；
- R8~R14是根据工作模式进行分组的寄存器；
- R15是程序计数器，也是不分组的；
- CPSR是当前程序状态寄存器；
- SPSR是各工作模式下保留CPSR值的寄存器。

# 不同工作模式下的寄存器分组

图

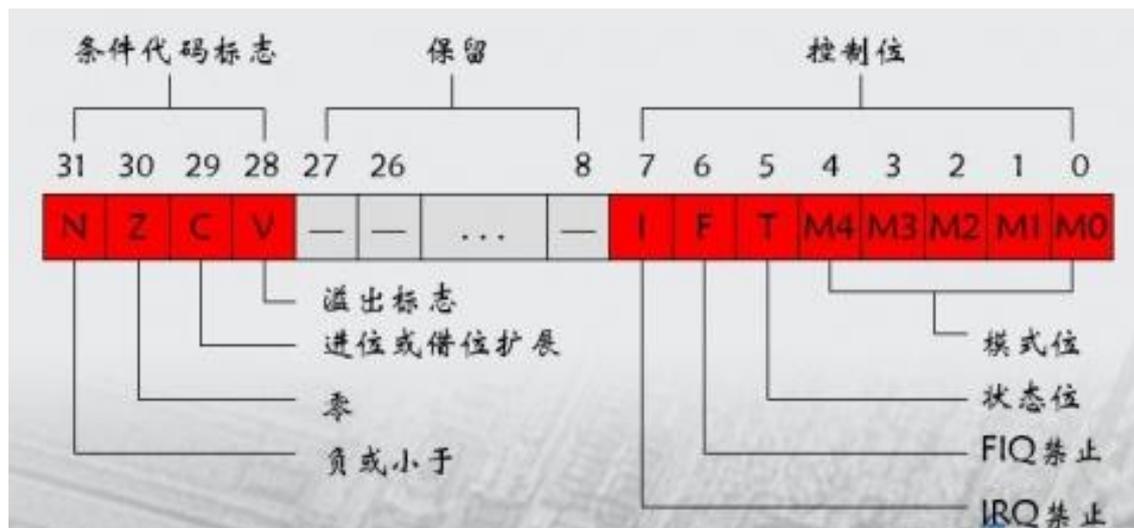
用户 <sup>⊥</sup>	系统 <sup>⊥</sup>	管理 <sup>⊥</sup>	中止 <sup>⊥</sup>	未定义 <sup>⊥</sup>	中断 <sup>⊥</sup>	快中断 <sup>⊥</sup>
R0 <sup>⊥</sup>	R0 <sup>⊥</sup>	R0 <sup>⊥</sup>	R0 <sup>⊥</sup>	R0 <sup>⊥</sup>	R0 <sup>⊥</sup>	R0 <sup>⊥</sup>
R1 <sup>⊥</sup>	R1 <sup>⊥</sup>	R1 <sup>⊥</sup>	R1 <sup>⊥</sup>	R1 <sup>⊥</sup>	R1 <sup>⊥</sup>	R1 <sup>⊥</sup>
R2 <sup>⊥</sup>	R2 <sup>⊥</sup>	R2 <sup>⊥</sup>	R2 <sup>⊥</sup>	R2 <sup>⊥</sup>	R2 <sup>⊥</sup>	R2 <sup>⊥</sup>
R3 <sup>⊥</sup>	R3 <sup>⊥</sup>	R3 <sup>⊥</sup>	R3 <sup>⊥</sup>	R3 <sup>⊥</sup>	R3 <sup>⊥</sup>	R3 <sup>⊥</sup>
R4 <sup>⊥</sup>	R4 <sup>⊥</sup>	R4 <sup>⊥</sup>	R4 <sup>⊥</sup>	R4 <sup>⊥</sup>	R4 <sup>⊥</sup>	R4 <sup>⊥</sup>
R5 <sup>⊥</sup>	R5 <sup>⊥</sup>	R5 <sup>⊥</sup>	R5 <sup>⊥</sup>	R5 <sup>⊥</sup>	R5 <sup>⊥</sup>	R5 <sup>⊥</sup>
R6 <sup>⊥</sup>	R6 <sup>⊥</sup>	R6 <sup>⊥</sup>	R6 <sup>⊥</sup>	R6 <sup>⊥</sup>	R6 <sup>⊥</sup>	R6 <sup>⊥</sup>
R7 <sup>⊥</sup>	R7 <sup>⊥</sup>	R7 <sup>⊥</sup>	R7 <sup>⊥</sup>	R7 <sup>⊥</sup>	R7 <sup>⊥</sup>	R7 <sup>⊥</sup>
R8 <sup>⊥</sup>	R8 <sup>⊥</sup>	R8 <sup>⊥</sup>	R8 <sup>⊥</sup>	R8 <sup>⊥</sup>	R8 <sup>⊥</sup>	R8_fiq* <sup>⊥</sup>
R9 <sup>⊥</sup>	R9 <sup>⊥</sup>	R9 <sup>⊥</sup>	R9 <sup>⊥</sup>	R9 <sup>⊥</sup>	R9 <sup>⊥</sup>	R9_fiq* <sup>⊥</sup>
R10 <sup>⊥</sup>	R10 <sup>⊥</sup>	R10 <sup>⊥</sup>	R10 <sup>⊥</sup>	R10 <sup>⊥</sup>	R10 <sup>⊥</sup>	R10_fiq* <sup>⊥</sup>
R11 <sup>⊥</sup>	R11 <sup>⊥</sup>	R11 <sup>⊥</sup>	R11 <sup>⊥</sup>	R11 <sup>⊥</sup>	R11 <sup>⊥</sup>	R11_fiq* <sup>⊥</sup>
R12 <sup>⊥</sup>	R12 <sup>⊥</sup>	R12 <sup>⊥</sup>	R12 <sup>⊥</sup>	R12 <sup>⊥</sup>	R12 <sup>⊥</sup>	R12_fiq* <sup>⊥</sup>
R13 <sup>⊥</sup>	R13 <sup>⊥</sup>	R13_svc* <sup>⊥</sup>	R13_abt* <sup>⊥</sup>	R13_und* <sup>⊥</sup>	R13_irq* <sup>⊥</sup>	R13_fiq* <sup>⊥</sup>
R14 <sup>⊥</sup>	R14 <sup>⊥</sup>	R14_svc* <sup>⊥</sup>	R14_abt* <sup>⊥</sup>	R14_und* <sup>⊥</sup>	R14_irq* <sup>⊥</sup>	R14_fiq* <sup>⊥</sup>
R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>	R15(PC) <sup>⊥</sup>
CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>	CPSR <sup>⊥</sup>
<sup>⊥</sup>	<sup>⊥</sup>	SPSR_svc <sup>⊥</sup>	SPSR_abt <sup>⊥</sup>	SPSR_und <sup>⊥</sup>	SPSR_irq <sup>⊥</sup>	SPSR_fiq <sup>⊥</sup>

- R13寄存器的作用通常是堆栈指针，又称为SP。每种异常模式都有对应于该模式下的R13物理寄存器。R13寄存器在初始化时，应设置为指向本模式下分配的堆栈空间的入口地址。



❖ **R14**寄存器被称为链接寄存器，又称为**LR (Link Register)**寄存器，实际上就是用作子程序调用时、或异常引起的程序分枝时的返回链接寄存器。

- CPSR（Current Program Status Register）寄存器称为当前程序状态寄存器，又称为R16。
- 在所有处理器模式下，CPSR都是同一个物理寄存器，它保存了程序运行的当前状态。如下图。



- 条件标志包括N标志、Z标志、C标志和V标志，它们的具体含义如下：
  - 1、N标志，当指令执行结果是带符号的二进制补码时，若结果为负数，则N标志位置1；若结果为正数或0，则N标志位置0。
  - 2、Z标志，又称为零标志，当指令执行结果为0时，Z标志置1；否则Z标志置0。
  - 3、C标志，又称进位标志。若是加法指令以及比较指令CMN，当指令执行结果产生进位时，则C标志置1，否则C标志置0；若是减法指令以及比较指令CMP，当指令执行结果产生借位时，则C标志置0；否则C标志置1；若是带有移位操作的非加法/减法指令，C标志值为移出的最后一位的值；若指令是其他非加法/减法指令，C标志不会改变。
  - 4、V标志，又称溢出标志。若是加法或减法指令，当指令执行结果产生带符号溢出时，V标志置1；若是非加法/减法指令，V标志不会改变。

- CPSR寄存器的第7位~第0位分别是I、F、T和M[4: 0]，它们均用做控制位。
- I和F分别是IRQ异常和FIQ异常的禁止/允许位：
  - 当I=1时，禁止IRQ异常，否则I=0时允许IRQ异常；
  - 当F=1时，禁止FIQ异常，否则F=0时允许FIQ异常。
- T位是微处理器状态位：
  - 当T=0时，指示为ARM状态；
  - 当T=1时，指示Thumb状态。

- M4、M3、M2、M1、M0是工作模式位，它们决定了ARM9微处理器的工作模式，如下表：

表 2-3 模式位

M[4: 0]	模式	可访问的寄存器
10000	用户	PC、R14~R0、CPSR
10001	FIQ	PC、R14_fiq~R8_fiq、R7~R0、CPSR、SPSR_fiq
10010	IRQ	PC、R14_irq、R13_irq、R12~R0、CPSR、SPSR_irq
10011	管理	PC、R14_svc、R13_svc、R12~R0、CPSR、SPSR_svc
10111	中止	PC、R14_abt、R13_abt、R12~R0、CPSR、SPSR_abt
11011	未定义	PC、R14_und、R13_und、R12~R0、CPSR、SPSR_und
11111	系统	PC、R14~R0、CPSR

# ARM9的异常

- 所谓异常，指的是由内部或外部产生一个引起微处理器处理的事件，换句话说，也就是指正常的程序执行流程被暂时中断而引发的过程。
- ARM920T微处理器核有7种异常，如下表：

表 2-4 异常处理模式

异常名称	对应模式	正常向量	高地址向量
复位	管理	0x00000000	0xFFFF0000
未定义指令	未定义	0x00000004	0xFFFF0004
软件中断 (SWI)	管理	0x00000008	0xFFFF0008
预取中止 (取指令存储器中止)	中止	0x0000000C	0xFFFF000C
数据中止	中止	0x00000010	0xFFFF0010
IRQ (中断)	IRQ	0x00000018	0xFFFF0018
FIQ (快速中断)	FIQ	0x0000001C	0xFFFF001C

- 下面对7种异常作进一步的解释：
- （1）复位异常：当系统上电及按下复位按键时，产生复位异常。将从地址0x00000000处开始执行程序。
- （2）未定义指令异常：当ARM920T微处理器执行未定义的指令时，会产生未定义指令异常。微处理器的PC值将被赋予0x00000004，开始从此地址处执行指令。
- （3）软件中断异常：微处理器执行软件中断指令SWI即产生软件中断异常，微处理器的PC值被赋予0x00000008，将从此地址处开始执行指令。

- (4) 中止异常：中止异常通常是发生在ARM9微处理器对存储器访问失败时，在存储器访问周期内，ARM9微处理器会检查是否发生中止异常。中止异常通常包括两种类型：一种是指令预取中止，另一种是数据中止。
- (5) IRQ异常：IRQ异常是由外部中断信号引起的。异常发生后，微处理器的PC值将被赋予0x00000018，将从此地址处开始执行程序。
- (6) FIQ异常：FIQ异常也是由外部中断信号引起的。异常发生后，微处理器的PC值将被赋予0x0000001C，将从此地址处开始执行程序。

- 异常优先级：若多个异常在某一时刻同时出现，那么，ARM9微处理器将按照异常优先级的高低顺序处理。
- ARM9异常的优先级排列顺序如下表：

表 2-5 优先级排列顺序

优先级	异常	优先级	异常
1（最高）	复位	4	IRQ
2	数据中止	5	预取中止
3	FIQ	6（最低）	未定义指令，SWI

- 当异常处理完成后，ARM9微处理器需退出异常，返回原来断点处继续执行。
- 退出异常将进行以下操作：
  - 1、将保存在R14寄存器的值，再回送到PC中。回送指令见下表：
  - 2、再将SPSR寄存器的值回送CPSR寄存器中。
  - 3、对中断禁止位标志进行清除。

发生转移条件	LR 回送 Pc 的指令	转移时 R14 寄存器所保存的值	
		ARM 状态	Thumb 状态
子程序调用	MOVS PC, R14	当前指令地址+4	当前指令地址+2
软件中断异常	MOVS PC, R14	当前指令地址+4	当前指令地址+2
未定义异常	MOVS PC, R14	当前指令地址+4	当前指令地址+2
FIQ 异常	SUBS PC, R14,#4	当前指令地址+8	当前指令地址+6
IRQ 异常	SUBS PC, R14,#4	当前指令地址+8	当前指令地址+6
指令预取中止异常	SUBS PC, R14,#4	当前指令地址+8	当前指令地址+6
数据预取中止异常	SUBS PC, R14,#8	当前指令地址+12	当前指令地址+10
复位	——		

# ARM9寻址方式

- 所谓寻址方式是指：微处理器根据指令中给出的地址码字段来寻找操作数所存储的真实地址的方式。
- **ARM9微处理器的基本寻址方式如下：**
  - (1) 立即寻址
  - (2) 寄存器寻址
  - (3) 寄存器移位寻址：ARM9指令集中具有一种特殊的寻址方式，即寄存器移位寻址方式。
  - (4) 寄存器间接寻址
  - (5) 变址寻址：变址寻址方式是将基址寄存器的内容与指令中给出的地址偏移量相加，从而得到操作数有效地址的一种寻址方式。变址寻址方式又分为：基址加偏移量和基址加索引二种寻址方式。

- (6) 多寄存器寻址：这种寻址方式允许在一条指令中完成传送几个寄存器的值，即允许用一条指令实现16个寄存器的任何子集的传送操作。例如：

LDMIA R2, {R3, R5, R7}

; R3=[R2], R5=[R2+4], R7=[R2+8]

- (7) 堆栈寻址：堆栈是一种常用的、用于保存信息的存储区域。这个存储区域的存储顺序是“先进后出”或“后进先出”。堆栈寻址方式使用了一个专门的寄存器，即堆栈指针SP，SP初始时指向该存储区域的顶。
- (8) 块拷贝寻址：块拷贝寻址指令完成的也是一种多寄存器传送功能，块拷贝寻址指令可以把一个连续的存储器区域的数据，拷贝到另一存储区域。
- (9) 相对寻址：相对寻址方式是变址寻址方式的一种特例，其基地址寄存器是程序计数器PC，而不是通用的寄存器Rn。