

嵌入式与ARM体系结构

- ARM公司简介
- ARM公司产业链
- ARM微处理器的应用领域
- ARM处理器的特点
- ARM体系结构版本
- ARM处理器内核系列
- ARM编程模式
- ARM内部寄存器

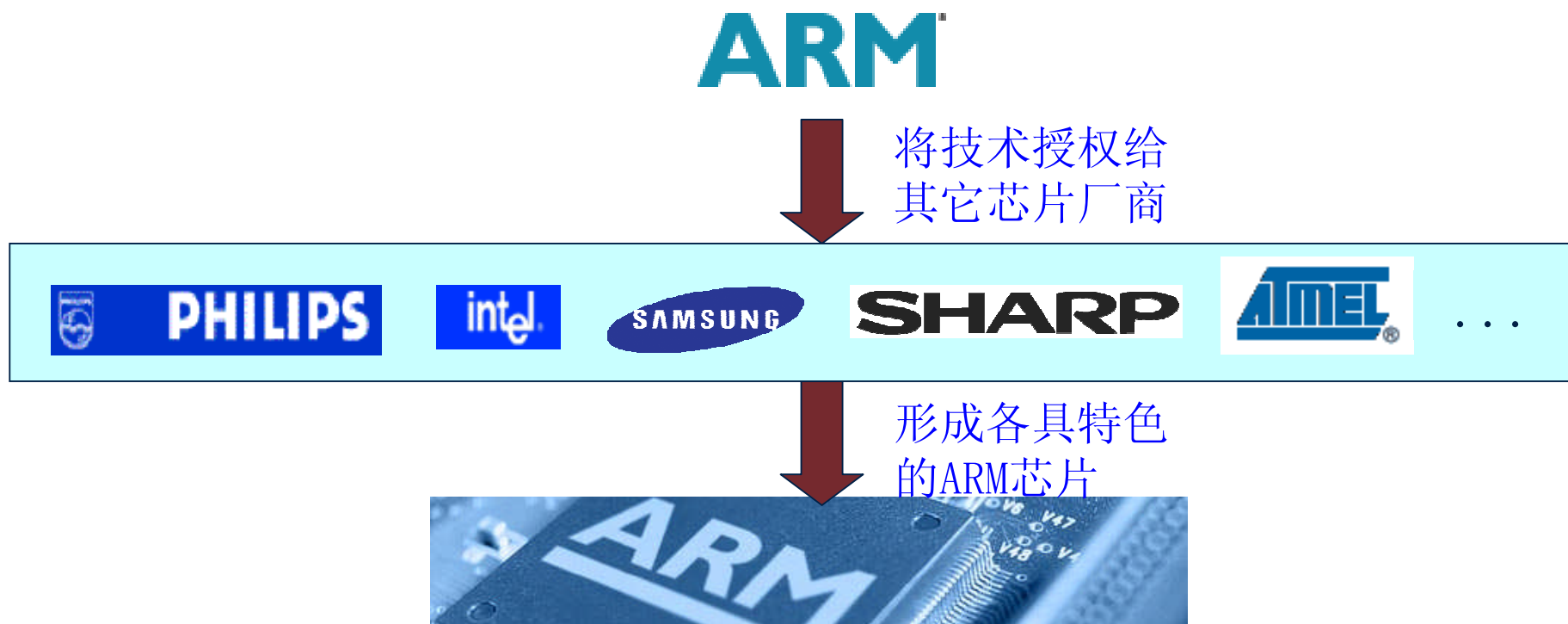
ARM公司简介

- 成立于1990年11月
 - 前身为 Acorn计算机公司
 - Advance RISC Machine (ARM)
- 主要设计ARM系列RISC处理器内核
- 授权ARM内核给生产和销售半导体的合作伙伴
 - ARM 公司不生产芯片
 - IP(Intelligence Property)
- 另外也提供基于ARM架构的开发设计技术
 - 软件工具, 评估板, 调试工具, 应用软件,
 - 总线架构, 外围设备单元, 等等





■ ARM公司产业链



■ ARM微处理器的应用领域

- 工业控制领域
- 无线通讯领域
- 网络应用
- 消费电子产品
- 成像和安全产品

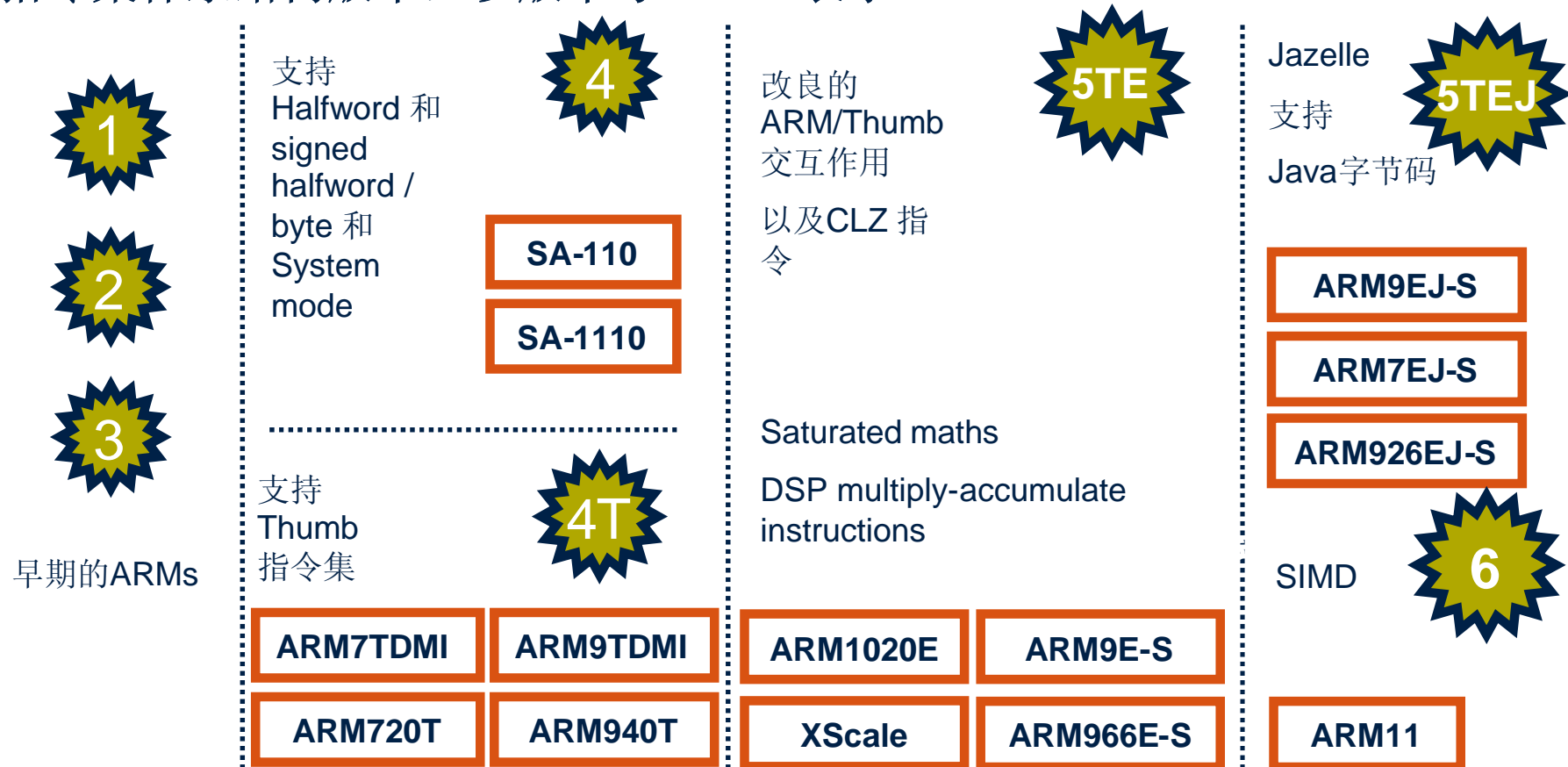
■ ARM处理器的特点

采用RISC架构的ARM微处理器一般具有如下特点：

- (1) 体积小、低功耗、低成本、高性能；
- (2) 支持Thumb（16位）/ARM（32位）双指令集，能很好的兼容8位/16位器件；
- (3) 大量使用寄存器，指令执行速度更快；
- (4) 大多数数据操作都在寄存器中完成；
- (5) 寻址方式灵活简单，执行效率高；
- (6) 指令长度固定。

ARM体系结构的版本

ARM体系结构从最初开发到现在有了巨大的改进，并仍在完善和发展。为了清楚的表达每个ARM应用实例所使用的指令集，ARM公司定义了6种主要的ARM指令集体系结构版本，以版本号V1~V6表示。



- ARM体系结构版本——V1
- 该版本的ARM体系结构，只有26位的寻址空间，没有商业化，其特点为：
 - 基本的数据处理指令（不包括乘法）；
 - 字节、字加载/存储指令（load/store）；
 - 具有分支指令，包括在子程序调用中使用的分支和链接指令；
 - 在操作系统调用中使用的软件中断指令。
 - 寻址空间：64M字节

• ARM体系结构版本——V2

同样为26位寻址空间，相对V1版本有以下改进：

- 具有乘法和乘加指令；
- 支持协处理器；
- 快速中断模式中的两个以上的分组寄存器；
- 具有原子性加载/存储指令SWP和SWPB（存储器与寄存器交换）。
- 寻址空间：64M字节

ARM体系结构版本——V3

将寻址范围扩展到32位（4G），增加了当前程序状态寄存器CPSR和程序状态寄存器保存寄存器SPSR以便于异常（Exception）的处理。增加了中止（Abort）和未定义二种处理器模式。指令集变化如下：

- （1）增加了两个指令MRS和MSR，允许访问新的CPSR和SPSR寄存器。
- （2）修改过去用于异常返回指令的功能以便继续使用。

- ARM体系结构版本——V4

不再为了与以前的版本兼容而支持26位体系结构，并明确了哪些指令会引起未定义指令异常发生，它相对V3版本作了以下的改进：

- 半字加载/存储指令；
- 字节和半字的加载和符号扩展（sign-extend）指令；
- 具有可以转换到Thumb状态的指令；
- 用户模式寄存器的新的特权处理器模式。
- 把一些未使用的指令空间捕获为未使用指令。

ARM体系结构版本——V5

在V4版本的基础上，对现在指令的定义进行了必要的修正，对V4版本的体系结构进行了扩展并增加了指令，具体如下：

- 改进了ARM/Thumb状态之间的切换效率；
- 增加计数前导零指令（CLZ）和软件断点指令（BRK）；
- 对乘法指令如何设置标志作了严格的定义；
- 为协处理器增加更多可选择的指令。

ARM体系结构版本——V6

在V5版本的基础上，对现在指令的定义进行了必要的修正，对V5版本的体系结构进行了扩展并增加了指令，具体如下：

- 增加了媒体指令（Media）

ARM体系结构总结

核	体系结构
ARM1	V1
ARM2	V2
ARM2aS, ARM3	V2a
ARM6, ARM600, ARM610	V3
ARM7, ARM700, ARM710	V3
ARM7TDMI, ARM710T, ARM720T, ARM740T	V4T
Strong ARM, ARM8, ARM810	V4
ARM9TDMI, ARM920T, ARM940T	V4T
ARM9E-S	V5TE
ARM10TDMI, ARM1020E	V5TE
ARM11	V6

版本2a是版本2的变种

在体系结构版本4的变种4T中，引入了16位Thumb压缩形式的指令集。

版本5TE在体系结构版本5T的基础上增加了信号处理指令集。

■ ARM微处理器系列

- ARM7 处理器内核系列
- ARM9 处理器内核系列
- ARM11 处理器内核系列
- 其他处理器

■ ARM7处理器系列

▶主要应用领域：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。

▶ ARM7系列微处理器包括如下几种类型的核：ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ。其中，ARM7TMDI是目前使用最广泛的32位嵌入式RISC处理器，属低端ARM处理器核。TDMI的基本含义为：

T：支持16位压缩指令集Thumb；

D：支持片上Debug；

M：内嵌硬件乘法器（Multiplier）

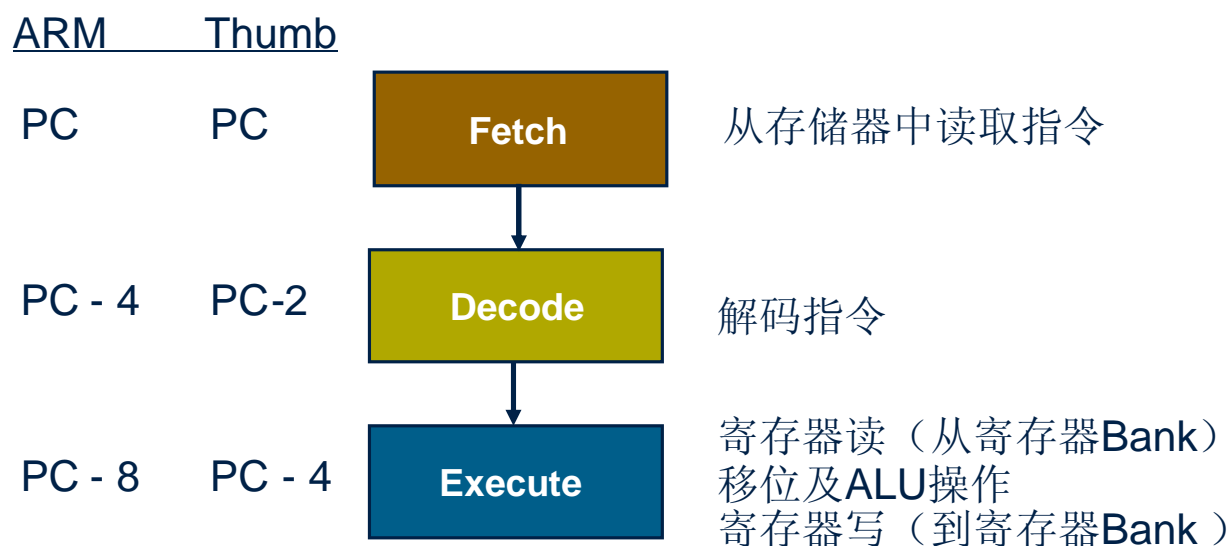
I：嵌入式ICE，支持片上断点和调试点；

■ ARM7TDMI 是基于 ARM7 内核

- 3 级流水线---0.9MIPS/MHz
- 冯. 诺依曼架构
- CPI (Cycle Per Instruction) 约为1.9
- T - Thumb 架构扩展, 提供两个独立的指令集:
 - ARM 指令, 均为 32位
 - Thumb指令, 均为 16位
 - 两种运行状态, 用来选择哪个指令集被执行
- D - 内核具有Debug扩展结构
- M - 增强乘法器 (32x8) 支持64位结果.
- I - EmbeddedICE-RT逻辑---提供片上断点和调试点支持

■ 指令流水线

- 为增加处理器指令流的速度，ARM7 系列使用3级流水线。
 - 允许多个操作同时处理，比逐条指令执行要快。



- PC指向正被取指的指令，而非正在执行的指令

最佳流水线

周期	1	2	3	4	5	6		
操作								
ADD	Fetch	Decode	Execute					
SUB		Fetch	Decode	Execute				
MOV			Fetch	Decode	Execute			
AND				Fetch	Decode	Execute		
ORR					Fetch	Decode	Execute	
EOR						Fetch	Decode	Execute
CMP							Fetch	Decode
RSB								Fetch

- 该例中用6个时钟周期执行了6条指令
- 所有的操作都在寄存器中（单周期执行）
- 指令周期数（CPI）= 1

■ ARM9处理器系列

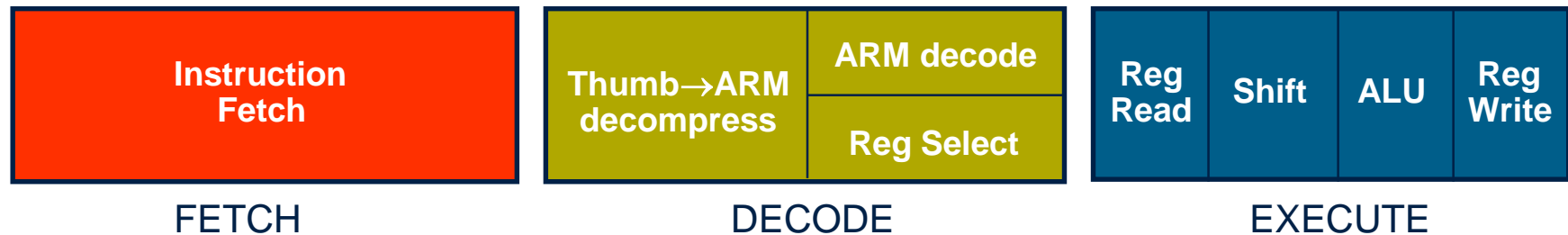
▶ ARM9系列微处理器主要应用于引擎管理、仪器仪表、安全系统、机顶盒、高端打印机、数字照相机和数字摄像机等。以及带mp3音频和MPEG4视频多媒体格式的智能电话。

▶ ARM9系列微处理器包含ARM920T、ARM922T和ARM940T三种类型，以适用于不同的应用场合。

- ARM9TDMI
- Harvard架构
 - 增加了可用的存储器宽度
 - 指令存储器接口
 - 数据存储器接口
 - 可以实现对指令和数据存储器的同时访问
- 5 级流水线
- 实现了以下改进:
 - 改进 CPI 到 ~ 1.5
 - 提高了最大时钟频率

ARM9TDMI流水线的变化

ARM7TDMI



ARM9TDMI



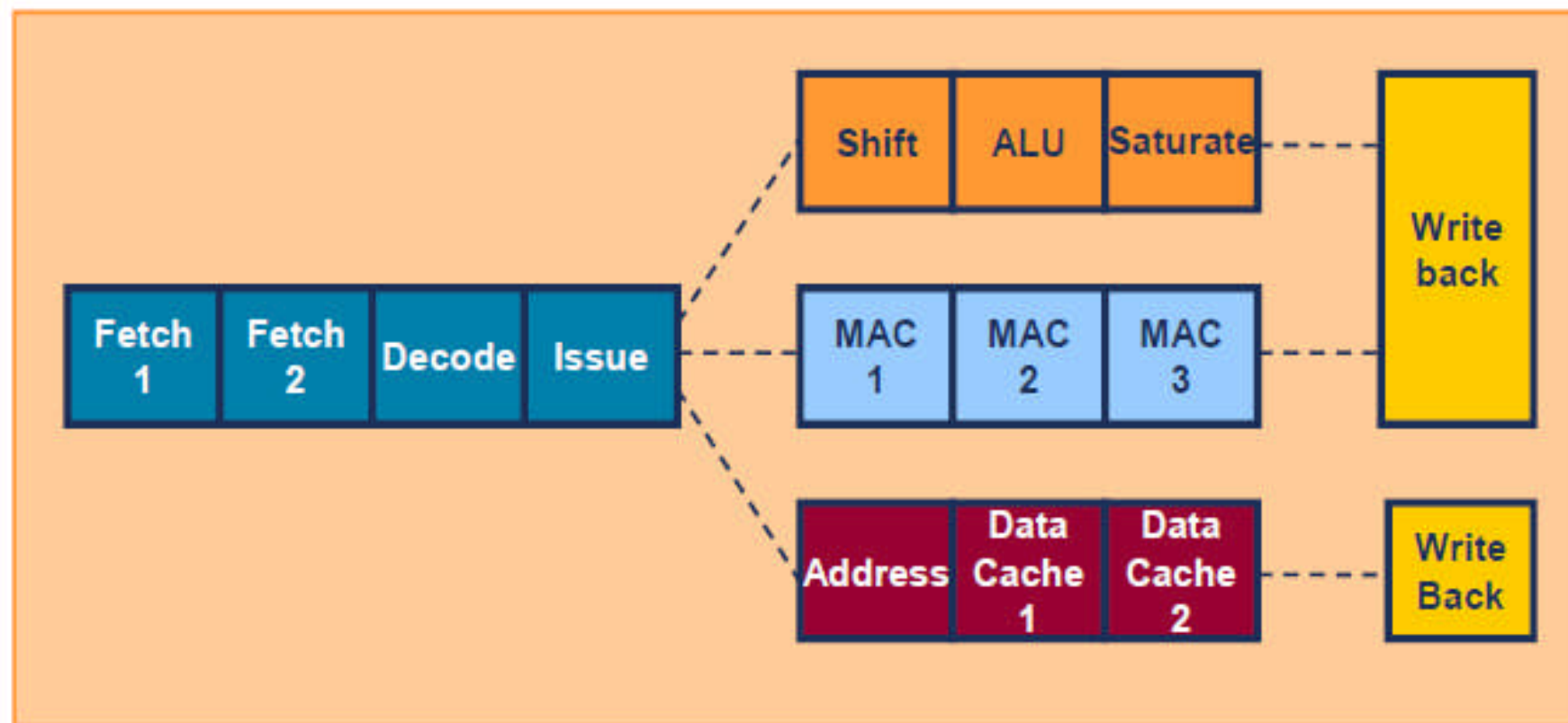
■ ARM11 处理器系列

- **ARM11™** 处理器系列所提供的引擎可用于当前生产领域中的很多智能手机；该系列还广泛用于消费类、家庭和嵌入式应用领域。
- 该处理器的功耗非常低，提供的性能范围为小面积设计中的 **350 MHz** 到速度优化设计中的 **1 GHz**（**45 纳米**和 **65 纳米**）。
- **ARM11** 处理器软件可以与以前所有 **ARM 处理器**兼容，并引入了用于媒体处理的 **32 位 SIMD**、用于提高操作系统上下文切换性能的物理标记高速缓存、强制实施硬件安全性的 TrustZone 以及针对实时应用的紧密耦合内存。

■ ARM11 处理器系列功能:

- 强大的 ARMv6 指令集体系结构
- ARM Thumb® 指令集可以减少高达 35% 的内存带宽和大小需求
- 用于执行高效嵌入式 Java 的 ARM Jazelle® 技术
- ARM DSP 扩展
- SIMD（单指令多数据）媒体处理扩展可提供高达 2 倍的视频处理性能
- 作为片上安全基础的 ARM TrustZone® 技术（ARM1176JZ-S 和 ARM1176JZF-S 处理器）
- Thumb-2 技术（仅 ARM1156(F)-S），可提高性能、能效和代码密度

ARM11 8级流水线



•ARM编程模式

ARM体系结构支持7种处理器模式，分别为：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，如下表所示。这样的好处是可以更好的支持操作系统并提高工作效率。ARM7TDMI完全支持这七种模式。

■ 处理器模式(如下表所示):

处理器模式	说明	备注
用户 (usr)	正常程序工作模式	不能直接切换到其它模式
系统 (sys)	用于支持操作系统的特权任务等	与用户模式类似, 但具有可以直接切换到其它模式等特权
快中断 (fiq)	支持高速数据传输及通道处理	FIQ 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/或存储器保护	在ARM7TDMI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

(1) 特权模式

处理器模式	说明	备注
用户 (usr)	正常程序工作模式	不能直接切换到其它模式
系统 (sys)	<p>除用户模式外，其它模式均为特权模式。ARM内部寄存器和一些片内外设在硬件设计上只允许（或者可选为只允许）特权模式下访问。此外，特权模式可以自由的切换处理器模式，而用户模式不能直接切换到别的模式。</p>	
快中断 (fiq)		
中断 (irq)		
管理 (svc)		
中止 (abt)		
未定义 (und)		

(2)异常模式

处理器模式	说明	备注
用户 (usr)	正常程序工作模式	不能直接切换到其它模式
系统 (sys)	<p>这五种模式称为异常模式。它们除了可以通过程序切换进入外，也可以由特定的异常进入。当特定的异常出现时，处理器进入相应的模式。每种异常模式都有一些独立的寄存器，以避免异常退出时用户模式的状态不可靠。</p>	
快中断 (fiq)		
中断 (irq)		
管理 (svc)		
中止 (abt)		
未定义 (und)		

(3)用户和系统模式

处理器模式	说明	备注
用户 (usr)	这两种模式都不能由异常进入，而且它们使用完全相同的寄存器组。	
系统 (sys)		
快中断 (fiq)		
中断 (irq)		
管理 (svc)		
中止 (abt)		
未定义 (und)	支持软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

这两种模式都不能由异常进入，而且它们使用完全相同的寄存器组。

系统模式是特权模式，不受用户模式的限制。操作系统在该模式下访问用户模式的寄存器就比较方便，而且操作系统的一些特权任务可以使用这个模式访问一些受控的资源。

■ 内部寄存器

- 在**ARM7TDMI**处理器内部有**37**个用户可见的寄存器。其中,**31**个通用**32**位寄存器及**6**个状态寄存器.
- 在不同的工作模式和处理器状态下, 程序员可以访问的寄存器也不尽相同。

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	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)	R6						
	R7(v4)	R7						
	R8(v5)	R8						R8_fiq
	R9(SB,v6)	R9						R9_fiq
	R10(SL,v7)	R10						R10_fiq
	R11(FP,v8)	R11						R11_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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

所有的37个寄存器，分成两大类：

■ 31个通用32位寄存器；

■ 6个状态寄存器。

ARM状态各模式下可以访问的寄存器

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)	R6						
	R7(v4)	R7						
	R8(v5)	R8						R8_fiq
	R9(SB,v6)	R9						R9_fiq
	R10(SL,v7)	R10						R10_fiq
	R11(FP,v8)	R11						R11_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

一般的通用寄存器

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		用户	系统	管理	中止	未定义	中断	快中断
通用寄存器	R0(a1)	R0						
	R1(a2)	R1						
		R2						
		R3						
		R4						
		R5						
		R6						
		R7						
		R8						R8_fiq
		R9						R9_fiq
		R10						R10_fiq
		R11						R11_fiq
		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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

在汇编语言中寄存器R0~R13为保存数据或地址值的通用寄存器。它们是完全通用的寄存器，不会被体系结构作为特殊用途，并且可用于任何使用通用寄存器的指令。

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		用户	系统	管理	中止	未定义	中断	快中断
通用寄存器		R0						
		R1						
		R2						
		R3						
		R4						
		R5						
		R6						
		R7						
	R8(V5)	R8						R8_fiq
	R9(SB,v6)	R9						R9_fiq
	R10(SL,v7)	R10						R10_fiq
	R11(FP,v8)	R11						R11_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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq	

其中R0~R7为未分组的寄存器，也就是说对于任何处理器模式，这些寄存器都对应于相同的32位物理寄存器。

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		用户	系统	管理	中止	未定义	中断	快中断
	R0(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5							
	R6							
	R7							
	R8						R8_fiq	
	R9						R9_fiq	
		R10						R10_fiq
		R11						R11_fiq
		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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq	

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

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)							
	R3(a4)							
	R4(v1)							
	R5(v2)							
	R6(v3)							
	R7(v4)							
	R8(v5)	R8						R8_fiq
	R9(SB,v6)	R9						R9_fiq
	R10(SL,v7)	R10						R10_fiq
	R11(FP,v8)	R11						R11_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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

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，另一个用于FIQ模式。这样在发生FIQ中断后，可以加速FIQ的处理速度。

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)	R6						
	R7(v4)	R7						
	R8(v5)	寄存器R13、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。						R8_fiq
	R9(SB,v6)							R9_fiq
	R10(SL,v7)							R10_fiq
	R11(FP,v8)							R11_fiq
	R12(IP)							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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)							
	R7(v4)							
	R8(v5)							R8_fiq
	R9(SB,v6)							R9_fiq
	R10(SL,v7)							R10_fiq
	R11(FP,v8)							R11_fiq
	R12(IP)							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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

寄存器R13常作为堆栈指针（SP）。在ARM指令集当中，没有以特殊方式使用R13的指令或其它功能，只是习惯上都这样使用。但是在Thumb指令集中存在使用R13的指令。

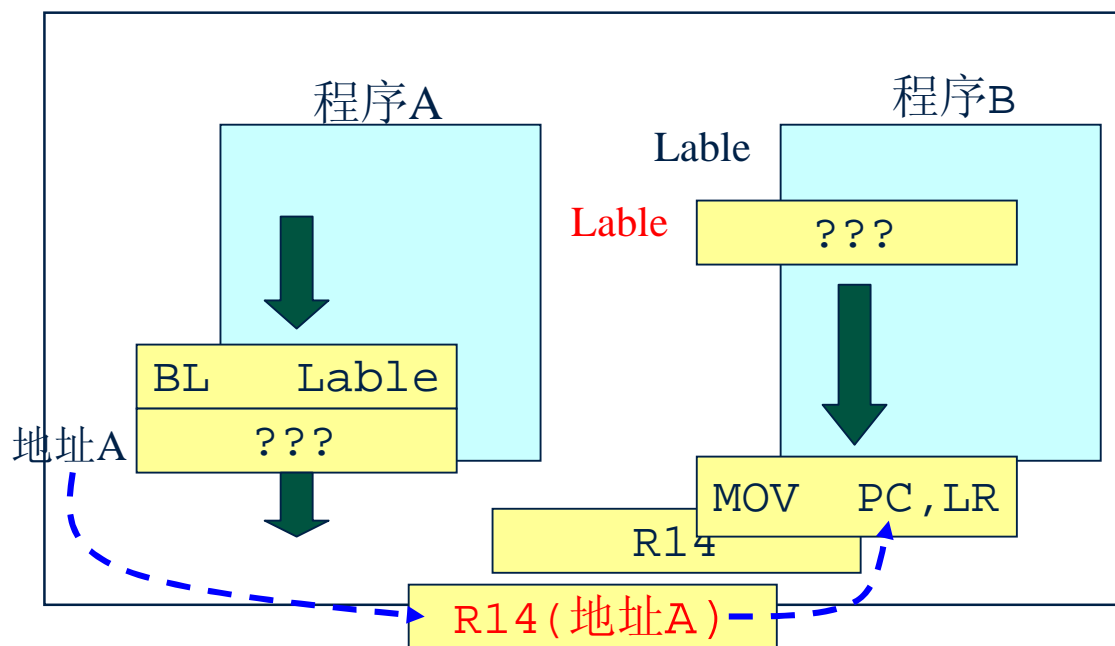
寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		用户	系统	管理	中止	未定义	中断	快中断
通用寄存器和程序计数器	R0(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)							
	R4(v1)							
	R5(v2)							
	R6(v3)							
	R7(v4)							
	R8(v5)							
	R9(SB,v6)							
	R10(SL,v7)							
	R11(FP,v8)							
	R12(IP)							
	R13(SP)	R13		R13_svc	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_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

R14为**链接寄存器**（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；
- 当发生异常时，将R14对应的异常模式版本设置为异常返回地址（有些异常有一个小的固定偏移量）。

■ R14寄存器与子程序调用

1. 程序A执行过程中调用程序B;
2. 程序跳转至标号Lable, 执行程序B。同时硬件将“BL Lable”指令的下一条指令所在地址存入R14;
3. 程序B执行最后, 将R14寄存器的内容放入PC, 返回程序A;



寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)	<p>寄存器R15为程序计数器（PC），它指向正在取指的地址。可以认为它是一个通用寄存器，但是对于它的使用有许多与指令相关的限制或特殊情况。如果R15使用的方式超出了这些限制，那么结果将是不可预测的。</p>						
	R7(v4)							
	R8(v5)							
	R9(SB,v6)							
	R10(SL,v7)							
	R11(FP,v8)							
	R12(IP)							
	R13(SP)							
	R14(LR)							
	R15(PC)	R15						
状态寄存器	CPSR	CPSR						
	SPSR	无		SPSR_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq

寄存器类别	寄存器在汇编中的名称	各模式下实际访问的寄存器						
		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(a1)	R0						
	R1(a2)	R1						
	R2(a3)	R2						
	R3(a4)	R3						
	R4(v1)	R4						
	R5(v2)	R5						
	R6(v3)							
	R7(v4)							
	R8(v5)							
	R9(SB,v6)							
	R10(SL,v7)							
	R11(FP,v8)							
	R12(IP)							
	R13(SP)							
	R14(LR)							
	R15(PC)	R15						
状态寄存器	CPSR	CPSR						
	SPSR	无	SPSR_abt	SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq	

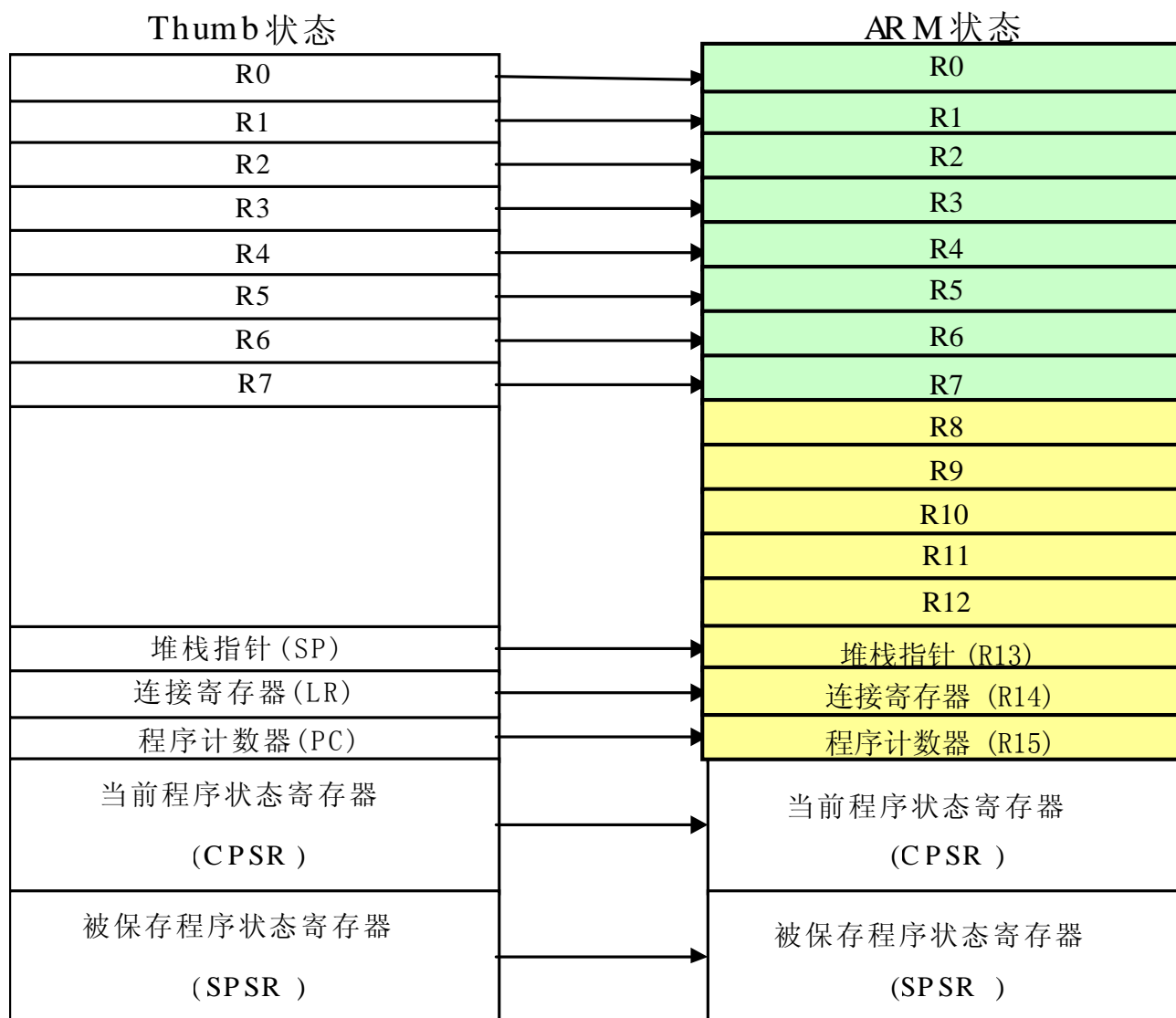
寄存器CPSR为**程序状态寄存器**，在异常模式中，另外一个寄存器“程序状态保存寄存器（SPSR）”可以被访问。每种异常都有自己的SPSR，在因为异常事件而进入异常时它保存CPSR的当前值，异常退出时可通过它恢复CPSR。详细描述以后还会讲。

■ Thumb状态寄存器

Thumb状态寄存器集是ARM状态集的子集，程序员可以直接访问的寄存器为：

- 8个通用寄存器R0～R7；
- 程序计数器（PC）；
- 堆栈指针（SP）；
- 链接寄存器（LR）；
- 有条件访问程序状态寄存器（CPSR）。

Thumb状态寄存器在Arm状态寄存器上的映射



■ 在Thumb状态中访问高寄存器

在Thumb状态中，高寄存器（R8～R15）不是标准寄存器集的一部分。汇编语言程序员对它们的访问受到限制，但可以将它们用于快速暂存。

可以使用MOV、CMP和ADD指令对高寄存器操作。

■ 程序状态寄存器

ARM7TDMI内核包含1个CPSR和5个供异常处理程序使用的SPSR。
CPSR反映了当前处理器的状态，其包含：

- 4个条件代码标志（负(N)、零(Z)、进位(C)和溢出(V)）；
- 2个中断禁止位，分别控制一种类型的中断；
- 5个对当前处理器模式进行编码的位；
- 1个用于指示当前执行指令(ARM还是Thumb)的位。



(1)条件代码标志

各标志位的含义如下：

- N 运算结果的最高位反映在该标志位。对于有符号二进制补码，结果为负数时N=1，结果为正数或零时N=0；
- Z 指令结果为0时Z=1（通常表示比较结果“相等”），否则Z=0；



各标志位的含义如下：

- C 当进行加法运算(包括CMN指令)，并且最高位产生进位时C=1，否则C=0。当进行减法运算(包括CMP 指令)，并且最高位产生借位时C=0，否则C=1。对于结合移位操作的非加法/减法指令，C为从最高位最后移出的值，其它指令C通常不变；
- V当进行加法/减法运算，并且发生有符号溢出时V=1，否则V=0，其它指令V通常不变。

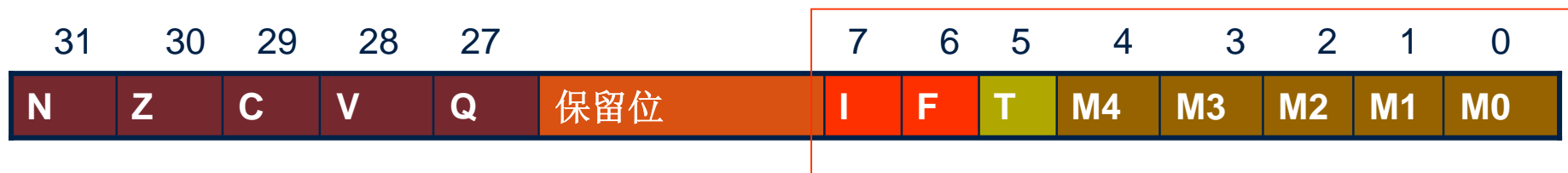


(2)控制位

CPSR的最低8位为控制位，当发生异常时，这些位被硬件改变。当处理器处于一个特权模式时，可用软件操作这些位。

它们分别是：

- 中断禁止位；
- T位；
- 模式位。



➤ 中断禁止位包括I和F位：

- 当I位置位时，IRQ中断被禁止；
- 当F位置位时，FIQ中断被禁止。

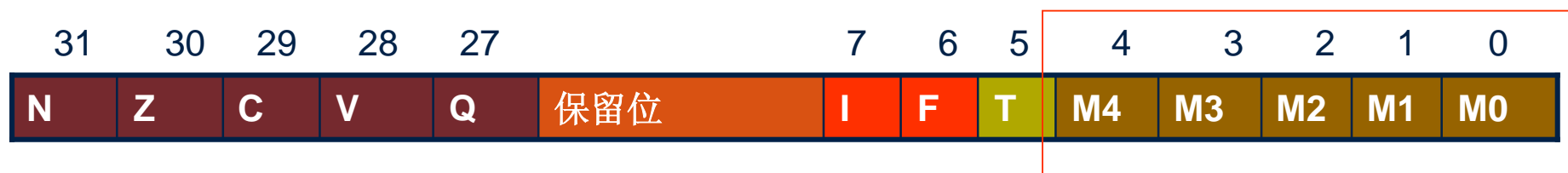
➤ T位反映了正在操作的状态：

- 当T位置位时，处理器正在Thumb状态下运行；
- 当T位清零时，处理器正在ARM状态下运行。



➤ 模式位包括M4、M3、M2、M1和M0，这些位决定处理器的操作模式。

注意：不是所有模式位的组合都定义了有效的处理器模式，如果使用了错误的设置，将引起一个无法恢复的错误。



•CPSR模式位设置表

M[4:0]	模式	可见的Thumb状态寄存器	可见的ARM状态寄存器
10000	用户	R0~R7,SP,LR,PC,CPSR	R0~R14,PC, CPSR
10001	快中断	R0~R7,SP_fiq,LR_fiq,PC,CPSR, SPSR_fiq	R0~R7,R8_fiq~R14_fiq,PC, CPSR, SPSR_fiq
10010	中断	R0~R7,SP_irq,LR_irq,PC,CPSR, SPSR_fiq	R0~R12,R13_irq,R14_irq,PC, CPSR, SPSR_irq
10011	管理	R0~R7,SP_svc,LR_svc,PC,CPSR, SPSR_svc	R0~R12,R13_svc,R14_svc, PC,CPSR, SPSR_svc
10111	中止	R0~R7,SP_abt,LR_abt,PC,CPSR, SPSR_abt	R0~R12,R13_abt,R14_abt,PC, CPSR, SPSR_abt
11011	未定义	R0~R7,SP_und,LR_und,PC,CPSR, SPSR_und	R0~R12, R13_und, R14_und, PC, CPSR, SPSR_und
11111	系统	R0~R7,SP,LR,PC,CPSR	R0~R14,PC, CPSR

简答题

1. 简述**ARM**微处理器的特点？
2. **ARM**微处理器系列包含多种微处理器，请简述其区别。
3. 请描述**ARM**寄存器用途及特点？
4. 简述**ARM**微处理器在**32**位机器上的寻址方式？
5. 说出**ARM**可以工作的模式名字。
6. **ARM**核有多少个寄存器？
7. **R13**通常用来存储什么？
8. 哪种模式使用的寄存器最少？
9. **ARM**有哪几个异常类型？

