

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

第二讲 ARM 基础知识

(此讲参考相关资料整理而成，难免存在不足之处，欢迎批评指正!)

2.1 ARM 简介

目前嵌入式处理器常见的有 ARM、PowerPC、MIPS、Motorola 68K、ColdFire(冷火)等，但 ARM 占据了绝对主流(有资料说手机中几乎 90%都是 ARM 处理器)。ARM 是一个只卖知识产权的公司，目前获得购买了 ARM CPU 核授权许可的大公司很多，包括 Intel、Samsung、Amstel、Motorola、Philip 等，他们都在 ARM CPU 核的基础上进行了一些外围扩展，形成自己的处理器。

ARM (Advanced RISC Machines)，既可以认为是一个公司的名字，也可以认为是对一类微处理器的通称，还可以认为是一种技术的名字。

1991 年 ARM 公司成立于英国剑桥，主要出售芯片设计技术的授权。目前，采用 ARM 技术知识产权(IP)核的微处理器，即我们通常所说的 ARM 微处理器，已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场，基于 ARM 技术的微处理器应用约占据了 32 位 RISC 微处理器 75%以上的市场份额，ARM 技术正在逐步渗入到我们生活的各个方面。

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，作为知识产权供应商，本身不直接从事芯片生产，靠转让设计许可由合作公司生产各具特色的芯片，世界各大半导体生产商从 ARM 公司购买 ARM 微处理器核，根据各自不同的应用领域，加入适当的外围电路，从而形成自己的 ARM 微处理器芯片进入市场。全世界有几十家大的半导体公司都使用 ARM 公司的授权，因此既使得 ARM 技术获得更多的第三方工具、制造、软件的支持，又使整个系统成本降低，使产品更容易进入市场被消费者所接受，更具有竞争力。(以上部分内容来源于互联网)

ARM 微处理器系列：

- ARM7 系列 (32 位低功耗 RISC 处理器，采用 ARMV4T [Newman] 结构，三级流水线)
- ARM9 系列 (采用 ARMV4T [Harvard] 结构，五级流水线)
- ARM9E 系列 (可综合处理器，支持 DSP 指令集)
- ARM10E 系列 (高性能、低功耗微处理器，采用 ARMV5 结构，6 级流水线)
- SecurCore 系列
- ARM11 系列

其中，ARM7、ARM9、ARM9E 和 ARM10E 为 4 个通用处理器系列。

[Http://hi.baidu.com/niejjinbo](http://hi.baidu.com/niejjinbo)

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

ARM 应用领域：

- 嵌入式控制（汽车、电子设备、保安设备、调制解调器、打印机等）
- 无线通信、便携式产品（手提式计算机、移动电话、PDA 等）
- 数字消费产品（数字音频播放器、数字机顶盒、数码相机、数字式电视机等）
- 据传，神 7 中也用到了 ARM 技术（ARM11），用到与否，有待考证

2.2 ARM7 微处理器系列简介

LPC2124 是 ARM7 内核的芯片，下面就先简单介绍下 ARM7

ARM7 系列微处理器为低功耗的 32 位 RISC 处理器，最适合用于对价位和功耗要求较高的消费类应用。ARM7 微处理器系列具有如下特点：

- 具有嵌入式 ICE - RT 逻辑，调试开发方便
- 极低的功耗，适合对功耗要求较高的应用，如便携式产品
- 能够提供 0.9MIPS/MHz 的三级流水线结构
- 代码密度高并兼容 16 位的 Thumb 指令集
- 对操作系统的支持广泛，包括 Windows CE、Linux、Palm OS 等
- 指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容，便于用户的产品升级换代
- 主频最高可达 130MIPS，高速的运算处理能力能胜任绝大多数的复杂应用

ARM7 系列微处理器的主要应用领域为：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，ARM7TDMI 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 为压缩指令集 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

在内核的基础上，生产厂商根据不同用户需求来配置生产 ARM 芯片，以满足不同市场需求，所以尽管使用相同的 ARM 内核，但不同的 ARM 功能可能相差很大。

基于 Proteus 的 ARM (LPC2124) 学习 ◆

作者：聂金波 2008 年 10 月 V1.0

2.3 ARM7 处理器运行模式

ARM7 微处理器支持 7 种运行模式：

- 用户模式 (User)
ARM 处理器正常的程序执行状态
- 快速中断模式 (FIQ)
用于高速数据传输或通道处理
- 外部中断模式 (IRQ)
用于通道的中断处理
- 管理模式 (Supervisor)
操作系统使用的保护模式
- 数据访问终止模式 (Abort)
当数据或指令预期终止时进入该模式，可用于虚拟存储及存储保护
- 系统模式 (System)
运行具有特权的操作系统任务
- 未定义指令终止模式 (Undefined)
当未定义的指令执行时进入该模式，可用于支持硬件协处理器的软件仿真

除用户模式 (User) 以外的其它六种模式都称为特权模式。除用户模式和系统模式 (System) 以外的其它 5 种模式又称为异常模式。

2.4 ARM7 寄存器

ARM 处理器共有 37 个 32 位寄存器，其中 31 个为通用寄存器，6 个为状态寄存器。具体分类如下：

- 31 个通用寄存器：
R0~R15;
R13_svc、R14_svc;
R13_abt、R14_abt;
R13_und、R14_und;
R13_irq、R14_irq;

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

R8_frq-R14_frq

- 6 个状态寄存器：

CPSR；SPSR_svc、SPSR_abt、SPSR_und、SPSR_irq 和 SPSR_fiq

通用寄存器又可以分为：

- (1) 未分组寄存器(The unbanked registers) R0-R7

R0-R7 是不分组寄存器。这意味着在所有处理器模式下，它们每一个都访问一样的 32 位寄存器。它们是真正的通用寄存器，没有体系结构所隐含的特殊用途。

- (2) 分组寄存器(The banked registers) R8 ~ R14

R8 ~ R14 是分组寄存器。它们每一个访问的物理寄存器取决于当前的处理器模式。若要访问特定的物理寄存器而不依赖当前的处理器模式，则使用规定的名字。

寄存器 R8 ~ R12 各有两组物理寄存器：一组为 FIQ 模式，另一组为除了 FIQ 以外的所有模式。寄存器 R8 ~ R12 没有任何指定的特殊用途。只是使用 R8 ~ R14 来简单地处理中断。寄存器 R13, R14 各有 6 个分组的物理寄存器。1 个用于用户模式和系统模式，其它 5 个分别用于 5 种异常模式。寄存器 R13 通常用做堆栈指针，称为 SP。每种异常模式都有自己的 R13。寄存器 R14 用作子程序链接寄存器，也称为 LR。

- (3) 程序计数器 R15 (PC)

寄存器 R15 用做程序计数器 (PC)。

程序状态寄存器

- (1) 当前程序状态寄存器 (CPSR Current Program status Register)

所有处理器模式下都可以访问当前的程序状态寄存器 CPSR。CPSR 包含条件码标志、中断禁止位、当前处理器模式以及其它状态和控制信息。

- (2) 备份的程序状态寄存器 (SPSR Saved Program status Register)

在每种异常模式下都有一个对应的物理寄存器——程序状态保存寄存器 SPSR。当异常出现时，SPSR 用于保存 CPSR 的状态，以便异常返回后恢复异常发生时的工作状态。

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

CPSR 的低 5 位用于定义当前操作模式，如下表：

CPSR[4: 0]	模式	用途
10000	用户	正常用户模式，程序正常执行模式
10001	FIQ	处理快速中断，支持高速数据传送或通道处理
10010	IRQ	处理普通中断
10011	SVC	操作系统保护模式，处理软件中断（SWI）
10111	终止	处理存储器故障，实现虚拟存储器和存储器保护
11011	未定义	处理未定义的指令陷阱，支持硬件协处理器的软件仿真
11111	系统	运行特权操作系统任务

表 2-1

程序状态寄存器的每一位安排如下：（从左到右 31: 0）

N	Z	C	V（保留位）	I	F	T	M4	M3	M2	M1	M0
---	---	---	---	------------	---	---	---	----	----	----	----	----

条件标志码：（Condition Code Flags）

N: Negative Z: Zero C: Carry V: OverFlow

控制位：（状态寄存器的低八位）

- 中断禁止位 I、F
I=1, 禁止 IRQ 中断；F=1, 禁止 FIQ 中断
- T 标志位：该位反映处理器的运行状态
T=1, Thumb 状态；T=0, ARM 状态
- 运行模式位（M[4: 0]）

见表 2-1

ARM 状态下的寄存器

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

寄存器类别		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	

图 2-1

Thumb 状态下的寄存器

Thumb 状态下的寄存器集是 ARM 状态下寄存器集的子集。程序员可以直接访问 8 个通用的寄存器 (R0~R7)，程序计数器 PC、堆栈指针 SP、连接寄存器 LR 和当前状态寄存器 CPSR。每一种特权模式都各有一组 SP，LR 和 SPSR。

- (1) Thumb 状态下和 ARM 状态下的 R0-R7 是相同的。
- (2) Thumb 状态下和 ARM 状态下的 CPSR 和所有的 SPSR 是相同的
- (3) Thumb 状态下的 SP 对应 ARM 状态下 R13
- (4) Thumb 状态下的 LR 对应 ARM 状态下 R14
- (5) Thumb 状态下的程序计数器 (PC) 对应 ARM 状态下 R15

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

2.5 ARM 存储格式

ARM 体系结构将存储器看作是从零地址开始的字节的线性组合。字节零到字节三放置第一个字 (WORD)，字节四到字节七存储第二个字，以此类推。

ARM 体系结构可以用两种方法存储字数据，分别称为大端格式和小端格式。

(1) 在大端格式中，字数据的高字节存储在低地址中，而字数据的低字节存储在高地址中，

如下图所示：

	31	23 22	15 14	8 7	0	
高地址	8	9	10	11		8
↑	4	5	6	7		4
低地址	0	1	2	3		0

表 2-2 以大端格式存放数据

(2) 与大端存储格式相反，在小端格式中，低地址中存放的是字数据的低字节，高地址存放的是字数据的高字节。

	31	23 22	15 14	8 7	0	
高地址	11	10	9	8		8
↑	7	6	5	4		4
低地址	3	2	1	0		0

表 2-3 以小端格式存放数据

2.6 ARM 流水线

ARM 流水线的执行分三个阶段：第一阶段持有从内存中取回的指令，第二阶段开始解码，而第三阶段实际执行它。所以，程序计数器总是超出当前执行指令的 2 个指令。

2.7 ARM 中断与异常

当正常的程序执行流程发生暂时的停止时，称之为异常。在处理异常之前，当前处理器的状态必须保留，这样当异常处理完之后，当前程序可以继续执行。处理器允许多个异常同时发生，他们将会按照固定的优先级进行执行。

ARM 内核支持 7 种中断，不同的中断处于不同的处理模式，具有不同的优先级，而且每个中断都有固定的中断入口地址，见下表：

[Http://hi.baidu.com/niejjinbo](http://hi.baidu.com/niejjinbo)

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

中断类型	处理模式	入口地址	优先级	中断返回指令
复位 (Reset)	Supervisor	0x00	1(最高)	
未定义指令(Undefine Instrction)	Undefine	0x04	6(最低)	MOVS pc, lr
软件中断 (Software Interrupt)	Supervisor	0x08	6	MOVS pc, lr
指令预取终止 (Prefech Abort)	Abort	0x0c	5	SUBS pc, lr, #4
数据终止 (DATA Abort)	Abort	0x10	2	SUBS pc, lr, #4
外部中断请求 (IRQ)	IRQ	0x18	4	SUBS pc, lr, #4
快速中断请求 (FIQ)	FIQ	0x1c	3	SUBS pc, lr, #4

表 2-4 ARM 内核的 7 种中断

Reset	当处理器的复位电平有效时，产生复位异常，程序跳转到复位异常处理程序执行
Undefine Instrction	当 ARM 处理器或协处理器遇到不能处理的指令时，产生未定义指令异常。可使用该异常机制进行软件仿真。
Software Interrupt	该异常由执行 SWI 指令产生，可用于用户模式下的程序调用特权操作指令。可使用该异常机制实现系统功能调用
Prefech Abort	若处理器预取的指令地址不存在，或该地址不允许当前指令访问时，存储器向处理器发出中止信号，但当预取的指令被执行时，才会产生指令预取中止异常
DATA Abort	若处理器数据访问指令的地址不存在，或该地址不允许当前指令访问时，产生数据中止异常
IRQ	当 ARM 外部中断请求引脚有效，且 CPSR 中的 I 位为 0 时，产生 IRQ 异常。系统的外设可通过该异常请求中断处理
FIQ	当 ARM 快速中断请求引脚有效，且 CPSR 中的 F 位为 0 时，产生 FIQ 异常

表 2-5 ARM 异常及具体含义

对异常的反应：

- (1) 将下一条指令的地址存入相应的链接寄存器 LR
- (2) 将 CPSR 复制到相应的 SPSR
- (3) 根据异常类型，强制设置 CPSR 的运行模式位
- (4) 强制 PC 从相关的异常向量地址取下一条指令执行，从而跳转到相应的异常处理程序处

[Http://hi.baidu.com/niejjinbo](http://hi.baidu.com/niejjinbo)

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

从异常返回：

- (1) 将链接寄存器 LR 的值减去相应的偏移量后送到 PC 中
- (2) 将 SPSR 复制回 CPSR 中
- (3) 若在进入异常处理时设置了中断禁止位，则要在此清楚

可以认为应用程序总是从复位异常处理程序开始执行的，因此复位异常处理程序不需要返回。

2.8 LPC2124 简介

LPC2124 是 PHILIPS 公司生产的单片 32 位 ARM 微控制器，是基于一个支持实时仿真和跟踪的 16/32 位 ARM7TDMI-S CPU，并带有 256KB 的嵌入的高速 FLASH 存储器。128 位宽度的存储器接口和独特的加速结构使 32 位代码能够在最大时钟速率下运行。对代码规模有严格控制的应用可使用 16 位 Thumb 模式将代码规模降低超过 30%，而性能的损失却很小。

由于 LPC2124 非常小的 64 脚封装、极低的功耗、多个 32 位定时器、4 路 10 位 ADC、PWM 输出、46 个 GPIO 以及多达 9 个外部中断使它们特别适用于工业控制、医疗系统、访问控制和电子收款机(POS)。由于内置了宽范围的串行通信接口，它们也非常适合于通信网关、协议转换器、嵌入式软件调制解调器以及其它各种类型的应用。

LPC2124 特性：

- 16/32 位 ARM7TDMI-S 核，超小 LQFP64 封装
- 16 kB 片内 SRAM
- 256 kB 片内 Flash 程序存储器，128 位宽度接口/加速器可实现高达 60 MHz 工作频率
- 通过片内 boot 装载程序实现在系统编程 (ISP) 和在应用编程 (IAP)
- 4 路 10 位 A/D 转换器，转换时间低至 2.44 μ s
- 2 个 32 位定时器(带 4 路捕获和 4 路比较通道)、PWM 单元(6 路输出)、实时时钟和看门狗
- 多个串行接口，包括 2 个 16C550 工业标准 UART、高速 I2C 接口(400 kHz)和 2 个 SPI 接口
- 通过片内 PLL 可实现最大为 60MHz 的 CPU 操作频率

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

- 向量中断控制器。可配置优先级和向量地址
- 多达 46 个通用 I/O 口(可承受 5V 电压)，9 个边沿或电平触发的外部中断引脚
- 2 个低功耗模式：空闲和掉电
- CPU 操作电压范围：1.65~1.95 V(1.8 V ± 0.15 V)
- I/O 操作电压范围：3.0~3.6 V(3.0 V ± 10%)，可承受 5V 电压

LPC2124 管脚图

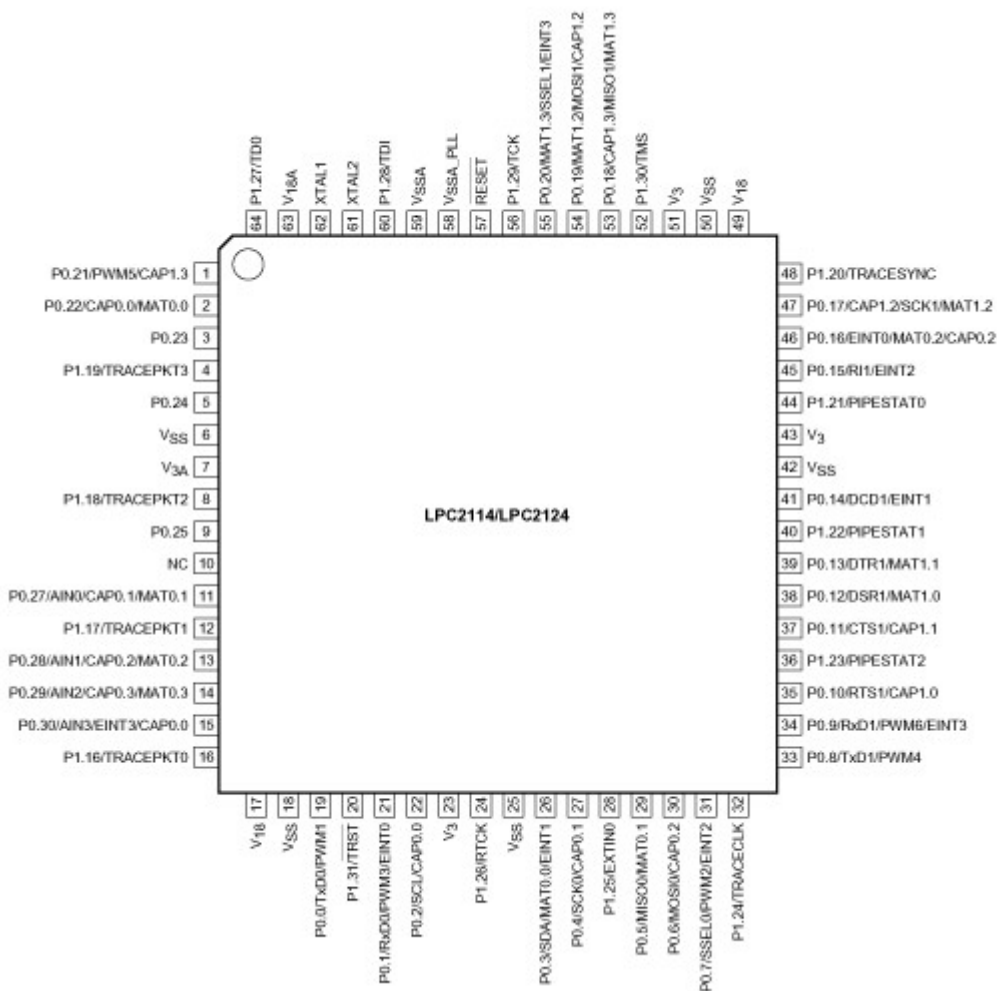


图 2-2

2.9 LPC2124 的管脚描述及其主要功能见下表：

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

管脚名称	LQF64 管脚#	类型	描述		
P0.0 P0.6		I/O	P0 口：P0 口是一个 32 位双向 I/O 口，每位的方向可单独控制。P0 口的功能取决于管脚连接模块的管脚功能选择。P0 口的 26 和 31 脚未用。 注：除用作 A/D 输入的管脚（P0.27, P0.28, P0.29 和 P0.30）外，所有 P0 管脚最大可承受 5V 的电压。如果未使用 A/D 转换器，A/D 输入可用作可承受 5V 电压的数字 I/O 口。见“A/D 转换器”。		
	19	0	P0.0	TxD0	UART0 发送输出端
		0		PWM1	脉宽调制器输出 1
	21	I	P0.1	RxD0	UART0 接收输入端
		0		PWM3	脉宽调制器输出 3
		I		EINT	外部中断 0 输入
	22	I/O	P0.2	SCL0	I2C0 时钟输入输出（开漏输出）
		I		CAP0.0	定时器 0 的捕获输入通道 0
	26	I/O	P0.3	SDA0	I2C0 数据输入输出（开漏输出）
		0		MAT0.0	定时器 0 的匹配输出通道 0
		I		EINT1	外部中断 1 输入
	27	I/O	P0.4	SCK0	SPI0 的串行时钟（SPI 时钟从主机输出，从机输入）
		I		CAP0.1	定时器 0 的捕获输入通道 0
	29	I/O	P0.5	MISO0	SPI0 主机输入从机输出端。从机到主机的数据传输。
		0		MAT0.1	定时器 0 的匹配输出通道 0
30	I/O	P0.6	MOSIO	SPI0 主机输出从机输入端，主机到从机的数据传输	
	I		CAP0.2		

[Http://hi.baidu.com/niejjinbo](http://hi.baidu.com/niejjinbo)

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

					定时器 0 的捕获输入通道 0
P0.7 P0.15	31	I	P0. 7	SSEL0	SPI0 从机选择。选择 SPI 接口用作从机。
		0		PWM2	脉宽调制器输出 2
		I		EINT2	外部中断 2 输入
	33	0	P0. 8	TxD1	UART1 发送输出端。
		0		PWM4	脉宽调制器输出 4
	34	I	P0. 9	RxD1	UART1 接收输入端。
		0		PWM6	脉宽调制器输出 6
		I		EINT3	外部中断 3 输入
	35	0	P0. 10	RTS1	UART1 请求发送输出端
		I		CAP1. 0	定时器 1 的捕获输入通道 0
	37	I	P0. 11	CTS1	UART1 清除发送输入端
		I		CAP1. 1	定时器 1 的捕获输入通道 1
	38	I	P0. 12	DSR1	UART1 数据设置就绪端
		0		MAT1. 0	定时器 1 的匹配输出通道 0
	39	0	P0. 13	DTR1	UART1 数据终止就绪端
		0		MAT1. 1	定时器 1 的匹配输出通道 1
	41	I	P0. 14	DCD1	UART1 数据载波检测输入端。
		0		EINT1	外部中断 1 输入。RESET 为低时，该管脚线上的低电平将强制片内引导装载程序复位后控制器件的操作。 重点：RESET 为低时，P0. 14 的低电平将强制片内引导装载程序复位后控制器件的操作
45	I	P0. 15	RI1	UART1 铃响指示输入端	
	I		EINT2	外部中断 2 输入	

[Http://hi.baidu.com/niejjinbo](http://hi.baidu.com/niejjinbo)

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

P0.16 P0.25	46	I	P0.16	EINT0	外部中断 0 输入
		0		MAT0.2	定时器 0 的匹配输出通道 2
		I		CAP0.2	定时器 0 的捕获输入通道 2
	47	I	P0.17	CAP1.2	定时器 1 的捕获输入通道 2
		I/O		SCK1	SPI1 串行时钟。SPI 时钟从主机输出或输入到从机
		0		MAT1.2	定时器 1 的匹配输出通道 2
	53	I	P0.18	CAP1.3	定时器 1 的捕获输入通道 3
		I/O		MISO1	SPI1 主机输入从机输出端。从机到主机的数据传输
		0		MAT1.3	定时器 1 的捕获输入通道 3
	54	0	P0.19	MAT1.2	定时器 1 的匹配输出通道 2
		I/O		MOSI1	SPI1 主机输出从机输入端。主机到从机的数据传输。
		0		CAP1.2	定时器 1 的捕获输入通道 2
	55	0	P0.20	MAT1.3	定时器 1 的捕获输入通道 3
		I		SSEL1	SPI1 从机选择。选择 SPI 接口用作从机
		I		EINT3	外部中断 3 输入
1	0	P0.21	PWM5	脉宽调制器输出 5。	
	I		CAP1.3	TIMER1 的捕获输入通道 3	
2	I	P0.22	CAP0.0	TIMER0 的捕获输入通道 0	
	0		MAT0.0	TIMER0 的匹配输出通道 0	
3	I/O	P0.23		通用双向数字端口	
5	I/O	P0.24		通用双向数字端口	
9	I/O	P0.25		通用双向数字端口	

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

P0.27 P0.30	11	I	P0. 27	AIN0	A/D 转换器输入 0。该模拟输入总是连接到相应的管脚上
		I		CAPO. 1	定时器 0 的捕获输入通道 1。
		0		MAT0. 1	定时器 0 的匹配输出通道 1
	13	I	P0. 28	AIN1	A/D 转换器输入 1。该模拟输入总是连接到相应的管脚上
		I		CAPO. 2	定时器 0 的捕获输入通道 2
		0		MAT0. 2	定时器 0 的匹配输出通道 2
	14	I	P0. 29	AIN2	A/D 转换器输入 2。该模拟输入总是连接到相应的管脚上。
		I		CAPO. 3	定时器 0 的捕获输入通道 3
		0		MAT0. 3	定时器 0 的匹配输出通道 3
	15	I	P0. 30	AIN3	A/D 转换器输入 3。该模拟输入总是连接到相应的管脚上
		I		EINT3	外部中断 3 输入
		0		CAPO. 0	定时器 0 的捕获输入通道 0
P1.16 P1.19		I/O	P1 口：P1 口是一个 32 位双向 I/O 口，每位的方向可单独控制。 P1 口的功能取决于管脚连接模块的管脚功能选择。P1 口只有 16 到 31 脚可用 注：P1 口是内置有上拉电阻的最大可承受 5V 电压的 I/O 口，用作输入口时可通过上拉电阻将输入电平置高		
16	0	P1. 16	TRACEPKT0	跟踪包位 0。带内部上拉的标准 I/O 口	
12	0	P1. 17	TRACEPKT1	跟踪包位 1。带内部上拉的标准 I/O 口	
8	0	P1. 18	TRACEPKT2	跟踪包位 2。带内部上拉的标准 I/O 口	

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

P1.20	4	0	P1.19	TRACEPKT3	跟踪包位 3。带内部上拉的标准 I/O 口
	48	0	P1.20	TRACESYNC	跟踪同步。标准 I/O 口带内部上拉。RESET 为低时，该管脚线上的低电平使 P1.25:16 复位后用作跟踪端口。 重点：RESET 为低时，P1.20 的低电平使 P1.25:16 复位后用作跟踪端口
P1.31	44	0	P1.21	PIPESTAT0	流水线状态位 0，带内部上拉的标准 I/O 口
	40	0	P1.22	PIPESTAT1	流水线状态位 1，带内部上拉的标准 I/O 口
	36	0	P1.23	PIPESTAT2	流水线状态位 2，带内部上拉的标准 I/O 口
	32	0	P1.24	TRACECLK	跟踪时钟。带内部上拉的标准 I/O 口
	28	I	P1.25	EXTIN0	外部触发输入。带内部上拉的标准 I/O 口
	24	I/O	P1.26	RTCK	返回的测试时钟输出。它是加载在 JTAG 接口的额外信号。辅助调试器与处理器频率的变化同步。双向管脚带内部上拉。RESET 为低时，该管脚线上的低电平使 P1.31:26 复位后用作一个调试端口。 重点：RESET 为低时，P1.26 的低电平使 P1.31:26 复位后用作一个调试端口
	64	0	P1.27	TDO	JTAG 接口的测试数据输出
	60	I	P1.28	TDI	JTAG 接口的测试数据输入
	56	I	P1.29	TCK	JTAG 接口的测试时钟
	52	I	P1.30	TMS	JTAG 接口的测试方式。
20	I	P1.31	/TRST	JTAG 接口的测试复位。	
NC	10	0	管脚悬空		

基于 Proteus 的 ARM (LPC2124) 学习

作者：聂金波 2008 年 10 月 V1.0

/RESET	57	I	外部复位输入： 当该管脚为低电平时，器件复位，I/O 口和外围功能进入默认状态，处理器从地址 0 开始执行程序。具有迟滞作用的 TTL 电平，管脚可承受 5V 电压
XTAL1	62	I	振荡器电路和内部时钟发生电路的输入
XTAL2	61	O	振荡放大器的输出
VSS	6, 18, 25, 42, 50	I	地： 0V 电压参考点。
VSSA	59	I	模拟地： 0V 电压参考点。它与 Vss 的电压相同，但为了降低噪声和出错几率，两者应当隔离
VSSA_PLL	58	I	PLL 模拟地： 0V 电压参考点。它与 Vss 的电压相同，但为了降低噪声和出错几率，两者应当隔离
V18	17, 49	I	1.8V 内核电源： 内部电路的电源电压。
V18A	63	I	模拟 1.8V 内核电源： 内部电路的电源电压。它与 V18 的电压相同，但为了降低噪声和出错几率，两者应当隔离
V3	24, 43, 51		3.3V 端口电源： I/O 口电源电压
V3A	7		模拟 3.3V 端口电源： 它与 V 但为了降低噪声和出错几率，两者应当隔离。 该管脚上的电平可用作 A/D 转换器的参考基准

走马观花，点到为止，ARM 基础知识部分就整理到这，水平有限，难免出现错误，欢迎批评指正！ARM 体系结构很复杂，仅靠小波整理的这点是远远不够的，建议想学 ARM 的朋友还是去花几十大洋去书店买本书。