

光盘附赠本书所有
实例源代码

AVR 单片机

应用开发指南 及实例精解

杨正忠 耿德根 编著



内容全面，知识性强

由浅入深地介绍了AVR单片机的基础知识、外围设备、主要功能和流行的单片机开发工具等知识点



重点突出，实用性强

以单片机工程项目开发的流程为主线，重点讲述了如何解决在实际编程中的问题，真正提高读者的应用开发能力

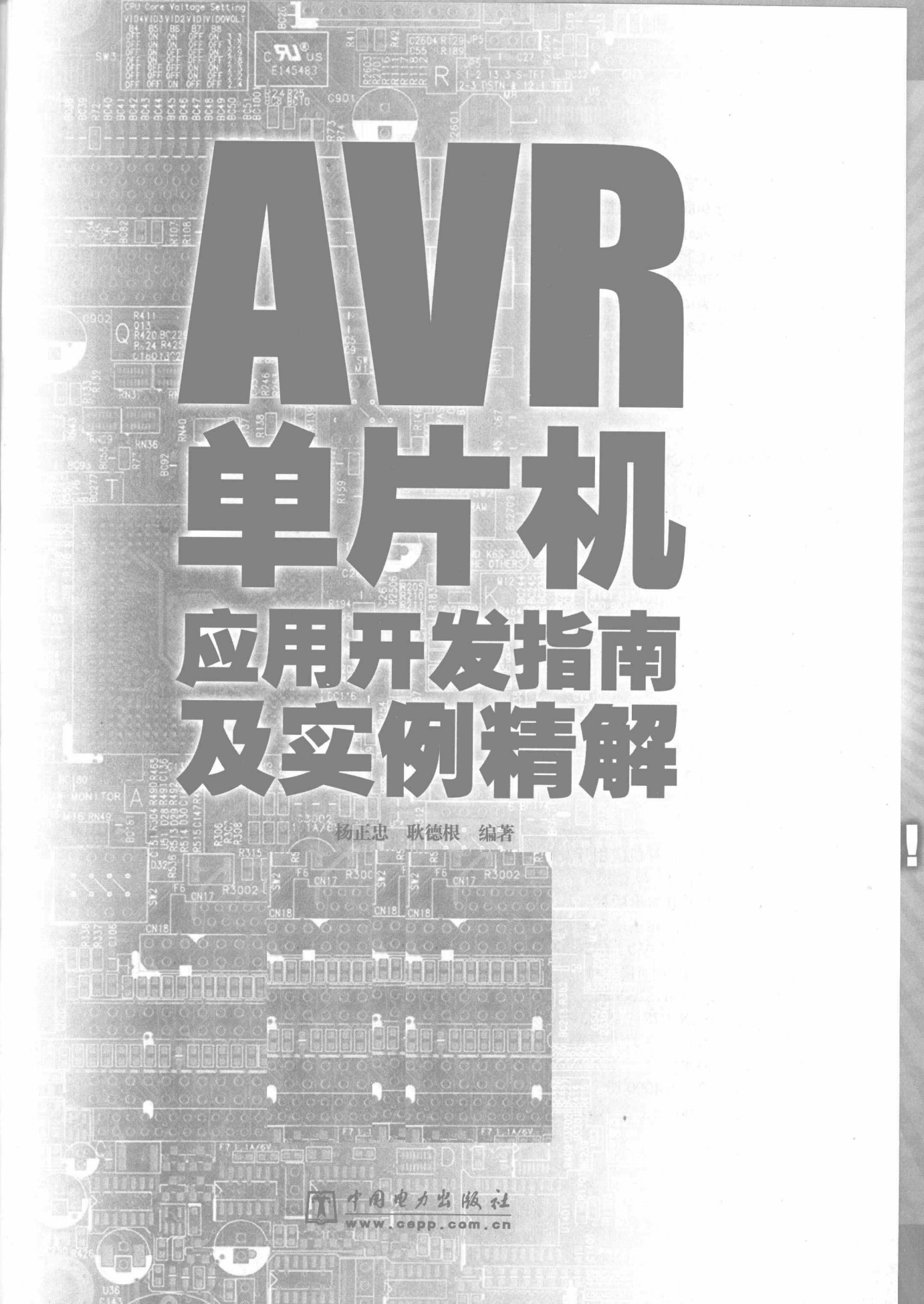


实例经典，指导性强

结合作者多年从事单片机产品研发的经验，结合大量精选的工程典型实例，让读者在实践中掌握单片机项目开发的应用技巧



中国电力出版社
www.cepp.com.cn



AVR

单片机

应用开发指南 及实例精解

杨正忠 耿德根 编著



中国电力出版社
www.cepp.com.cn

内 容 提 要

本书结构清晰,由浅入深,通俗易懂,结合 ATmega16/32 单片机阐述 AVR 单片机的理论和工程应用。

本书共分 9 章,主要包括以下内容:第 1 章介绍了 ATmega16/32 单片机的基础知识;第 2~6 章分别介绍了 ATmega16/32 单片机主要功能和外设的应用实例;第 7 章介绍了目前比较流行的 AVR 单片机开发工具;第 8 章介绍了 ATmega16/32 的综合应用;第 9 章通过温度检测与报警系统典型实例介绍了单片机开发项目流程和知识点应用。

本书可作为单片机初学者的学习用书,也可作为广大从事单片机应用系统开发的工程技术人员和相关人员的工作参考用书。

图书在版编目(CIP)数据

AVR 单片机应用开发指南及实例精解 / 杨正忠, 耿德根编著. —北京: 中国电力出版社, 2008.
ISBN 978-7-5083-7439-0

I. A… II. ①杨…②耿… III. 单片微型计算机 IV. TP368.1

中国版本图书馆 CIP 数据核字 (2008) 第 077260 号

责任编辑: 刘 炽

责任校对: 崔燕菊

责任印制: 郭华清

书 名: AVR 单片机应用开发指南及实例精解

编 著: 杨正忠 耿德根

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 航远印刷有限公司

开本尺寸: 185mm × 260mm 印 张: 21 字 数: 487 千字

书 号: ISBN 978-7-5083-7439-0

版 次: 2008 年 9 月北京第 1 版

印 次: 2008 年 9 月第 1 次印刷

印 数: 0001—4000 册

定 价: 38.00 元 (含 1CD)

敬 告 读 者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

现在单片机已经在很多领域得到了广泛的应用,但随着技术的进步和应用需求的不断发展,人们对处理器的综合性能要求也越来越高,也希望产品的性价比越来越高,特别是在一些中、低端应用领域,更期望通过集成度比较高,而且便宜的单片机来解决问题,主要体现在外设丰富、功能强大、能灵活运用、速度快、功耗低、产品开发方便、容易升级等方面。

ATMEL 公司推出的 AVR 系列单片机在功能、速度、功耗等方面具有独特的优势,而 ATmega 系列又属于该类型单片机的高档产品,其强大的功能、丰富的外设和低廉的价格正在吸引着更多的单片机工程师和相关从业人员。

由于 ATmega16/32 两款单片机在一般的 AVR 应用系统中比较常见,而且具有一定的代表性,同时目前关于这两款单片机的应用介绍比较少,为了使广大用户更快地掌握 ATmega16/32 的相关应用要点,笔者结合自己的体会和相关专业人士的指导,编写了此书。

本书共分为 9 章。第 1 章是 ATmega16/32 单片机介绍;第 2~6 章分别介绍了 ATmega16/32 主要功能和外设的应用说明;第 7 章介绍了目前比较流行的开发工具使用说明,如编译器和 SL-ISP 使用说明等;第 8 章介绍了 ATmega16/32 的综合应用实例。第 9 章通过一个实用案例,简要介绍了单片机开发项目流程和应用知识点。

由于编写本书的目的在于能从应用的角度让读者了解和理解 ATmega16/32 单片机,许多要点还可延伸到该系列的其他单片机,所以本书对一些知识点的原理介绍得不够充分,请读者根据需要参阅相关资料。本书可作为单片机初学者的学习用书,也可作为广大从事单片机应用系统开发的工程技术人员和相关人员的参考用书。

本书在编写过程中,一直得到了广州双龙电子公司的大力支持,为本书的编写提供了宝贵的建议,另外,韩国芳、谢鑫林、杨邦成、康华钰、邱祥永、果长红、马远、刘琦等人对本书的编写也提供了帮助,在此一并表示衷心的感谢。限于笔者的水平和经验,加之时间比较仓促,在编写过程中难免有不当或错误之处,敬请读者批评指正。

编者

2008 年 4 月

目 录

前 言

第 1 章 ATmega16/32 单片机介绍	1
1.1 ATmega16/32 单片机总体结构	1
1.2 ATmega16/32 单片机 CPU 结构	4
1.3 ATmega16/32 单片机存储器	8
1.4 ATmega16/32 单片机复位系统	11
1.5 ATmega16/32 单片机中断	11
1.6 ATmega16/32 单片机的省电方式	12
1.7 ATmega16/32 单片机定时器/计数器	14
1.8 ATmega16/32 单片机 TWI 接口说明	15
1.9 ATmega16/32 单片机内部 EEPROM	16
1.10 ATmega16/32 单片机通用串行接口	19
1.11 ATmega16/32 单片机模拟比较器	20
1.12 ATmega16/32 单片机 I/O 端口	21
1.13 ATmega16/32 单片机 AD 接口	27
1.14 ATmega16/32 单片机 SPI 接口	34
1.15 ATmega16/32 单片机看门狗说明	37
第 2 章 ATmega16/32 中断应用	39
2.1 ATmega16/32 中断资源说明	39
2.2 ATmega16/32 中断优先级	43
2.3 ATmega16/32 中断使用要点	43
2.4 ATmega16/32 中断应用实例	45
第 3 章 ATmega16/32 定时器/计数器应用	54
3.1 ATmega16/32 定时器/计数器详细说明	54
3.2 定时器/计数器操作范例	91
第 4 章 ATmega16/32 串行接口 USART 应用	103
4.1 ATmega16/32 USART 详细说明	103
4.2 USART 操作范例	115
4.3 USART 应用实例	119

第 5 章 ATmega16/32TWI 接口应用	123
5.1 ATmega16/32 TWI 接口详细说明	123
5.2 TWI 接口操作范例	131
5.3 TWI 应用实例	133
第 6 章 ATmega16/32 SPI 接口应用	139
6.1 ATmega16/32 SPI 接口详细说明	139
6.2 ATmega16/32 SPI 接口操作范例	144
6.3 SPI 接口应用实例	145
第 7 章 ATmega16/32 开发工具使用介绍	152
7.1 AVR Studio 使用介绍	152
7.2 ImageCraft ICCAVR 使用介绍	157
7.3 SL-ISP 下载软件使用介绍	170
第 8 章 ATmega16/32 综合应用	176
8.1 ATmega16/32 在短距离无线通信系统中的应用	176
8.2 ATmega16/32 在工业水务管网监测系统中的应用	184
8.3 ATmega16/32 在多功能饮水机中的应用	191
8.4 ATmega16/32 在数码音乐播放系统中的应用	203
8.5 ATmega16/32 在地质监测 GPRS/GSM 系统中的应用	220
8.6 ATmega16/32 在便携设备中的应用	239
第 9 章 低功耗温度检测与报警系统	254
9.1 系统概述	254
9.2 系统设计的准备工作	254
9.3 系统关键器件选择	255
9.4 系统设计硬件方案示意图	262
9.5 系统原理图示意图	262
9.6 系统工作流程示意图	269
9.7 系统部分程序介绍	270
9.8 小结	294
附录 A SL-DIY08-16 单片机实验器及教学机器人	295
附录 B ATmega16/32 头文件	298
附录 C AVR 汇编指令集	318
附录 D ASCII 字符对应表	322
参考文献	328



1.1 ATmega16/32 单片机总体结构

(1) ATmega16/32 单片机有如下特点:

①高性能、低功耗的 8 位 AVR 微处理器。

②先进的 RISC 结构。

- 131 条指令, 大多数指令执行时间为单个时钟周期。
- 32 个 8 位通用工作寄存器。
- 全静态工作。
- 工作于 16MHz 时性能高达 16 MIPS。
- 只需两个时钟周期的硬件乘法器。

③非易失性程序和数据存储器。

- 16KB (ATmega16) 或 32KB (ATmega32) 的系统内可编程 Flash, 擦写寿命为 10000 次。
- 具有独立锁定位的可选 Boot 代码区, 通过片上 Boot 程序实现系统内编程, 真正的同时读写操作。
- 512B (ATmega16) 或 1024B (ATmega32) 的 EEPROM, 擦写寿命为 100000 次。
- 1KB (ATmega16) 或 2KB (ATmega32) 的片内 SRAM。
- 可以对锁定位进行编程以实现用户程序的加密。

④JTAG 接口 (与 IEEE 1149.1 标准兼容)。

- 符合 JTAG 标准的边界扫描功能。
- 支持扩展的片内调试功能。
- 通过 JTAG 接口实现对 Flash、EEPROM、熔丝位和锁定位的编程。

⑤外设特点。

- 两个具有独立预分频器和比较器功能的 8 位定时器/计数器。
- 一个具有预分频器、比较功能和捕捉功能的 16 位定时器/计数器。
- 具有独立振荡器的实时计数器 RTC。
- 四通道 PWM。
- 8 路 10 位 ADC, 8 个单端通道, TQFP 封装的 7 个差分通道, 两个具有可编程增益 (1x、10x 或 200x) 的差分通道。
- 面向字节的两线接口。
- 两个可编程的串行 USART。
- 可工作于主机/从机模式的 SPI 串行接口。



- 具有独立片内振荡器的可编程看门狗定时器。
- 片内模拟比较器。
- ⑥特殊的处理器特点。
 - 上电复位以及可编程的掉电检测。
 - 片内经过标定的 RC 振荡器。
 - 片内/片外中断源。
 - 6 种睡眠模式，即空闲模式、ADC 噪声抑制模式、省电模式、掉电模式、Standby 模式，以及扩展的 Standby 模式。
- ⑦I/O 和封装。
 - 32 个可编程的 I/O 口。
 - 40 引脚 PDIP 封装，44 引脚 TQFP 封装，与 44 引脚 MLF 封装。
- ⑧工作电压。
 - ATmega16L, ATmega32L: 2.7~5.5V
 - ATmega16, ATmega32: 4.5~5.5V
- ⑨速度等级。
 - 0~8MHz ATmega16L (ATmega32L)。
 - 0~16MHz ATmega16 (ATmega32)。
- ⑩ATmega16L, ATmega32L 在 1MHz, 3V, 25℃ 时的功耗。
 - 正常模式为 1.1mA。
 - 空闲模式为 0.35mA。
 - 掉电模式时为小于 1μA。

(2) ATmega16/32 单片机的结构如图 1-1 所示。

通过结构图可以看出，ATmega16/32 是一类功能强大的单片机，可为许多嵌入式控制应用提供灵活而低成本的解决方案。

(3) ATmega16/32 单片机的引脚如图 1-2 所示。

(4) ATmega16/32 单片机引脚说明如表 1-1 所示。

表 1-1 ATmega16/32 单片机引脚说明

V _{CC}	数字电路的电源
GND	地
端口 A (PA7~PA0)	端口 A 作为 A/D 转换器的模拟输入端。端口 A 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。当作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 A 仍处于高阻状态
端口 B (PB7~PB0)	端口 B 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 B 仍处于高阻状态。端口 B 也可以用作其他不同的特殊功能，请参见数据手册
端口 C (PC7~PC0)	端口 C 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性，可以输出和吸收大电流。当作为输入使用时，若内部上拉电阻使能，端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 C 仍处于高阻状态。如果 JTAG 接口使能，即使复位出现引脚 PC5 (TDI)，PC3 (TMS) 与 PC2 (TCK) 的上拉电阻被激活。除去移出数据的 TAP 态外，TD0 引脚为高阻态。端口 C 也可以用作其他不同的特殊功能，请参见数据手册

续表

V_{CC}	数字电路的电源
端口 D (PD7~PD0)	端口 D 为 8 位双向 I/O 口，具有可编程的内部上拉电阻。其输出缓冲器具有对称的驱动特性。可以输出和吸收大电流。当作为输入使用时，若内部上拉电阻使能，则端口被外部电路拉低时将输出电流。在复位过程中，即使系统时钟还未起振，端口 D 仍处于高阻状态，端口 D 也可以用作其他不同的特殊功能
RESET	复位输入引脚。持续时间超过最小门限时间的低电平将引起系统复位。门限时间请参见数据手册
XTAL1	反向振荡放大器与片内时钟操作电路的输入端
XTAL2	反向振荡放大器的输出端
AVCC	AVCC 是端口 A 与 A/D 转换器的电源。当不使用 ADC 时，该引脚应直接与 V_{CC} 连接；当使用 ADC 时，应通过一个低通滤波器与 V_{CC} 连接
AREF	A/D 的模拟基准输入引脚

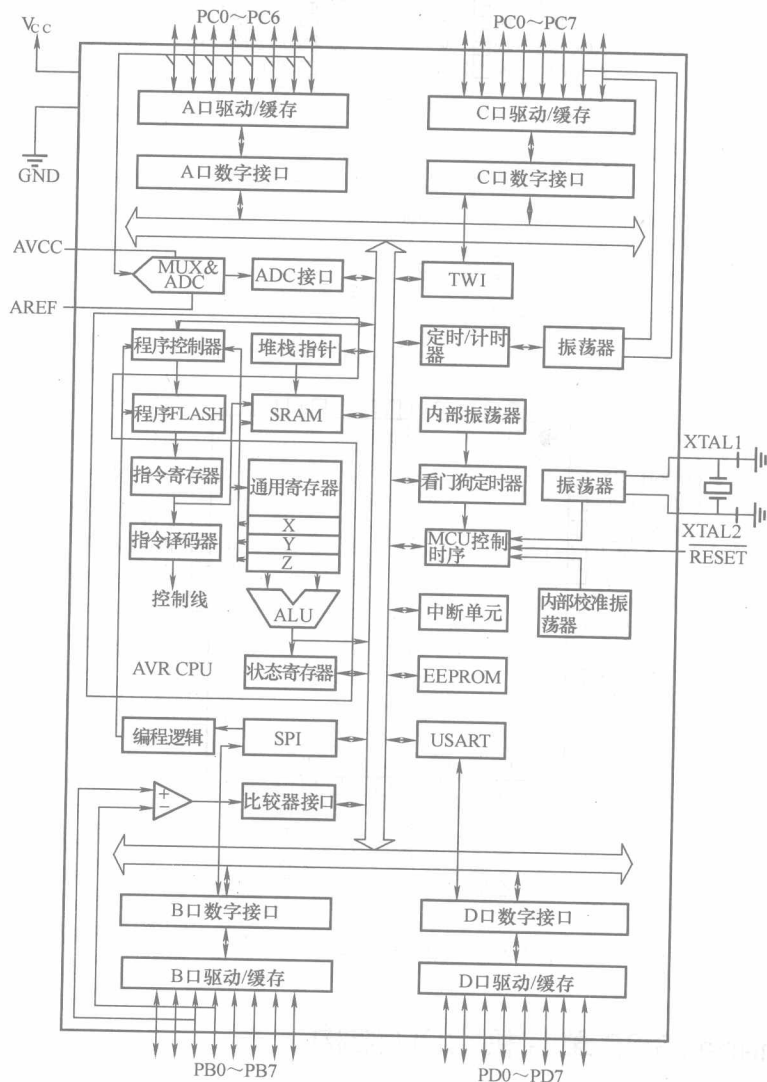


图 1-1 ATmega16/32 单片机结构框图

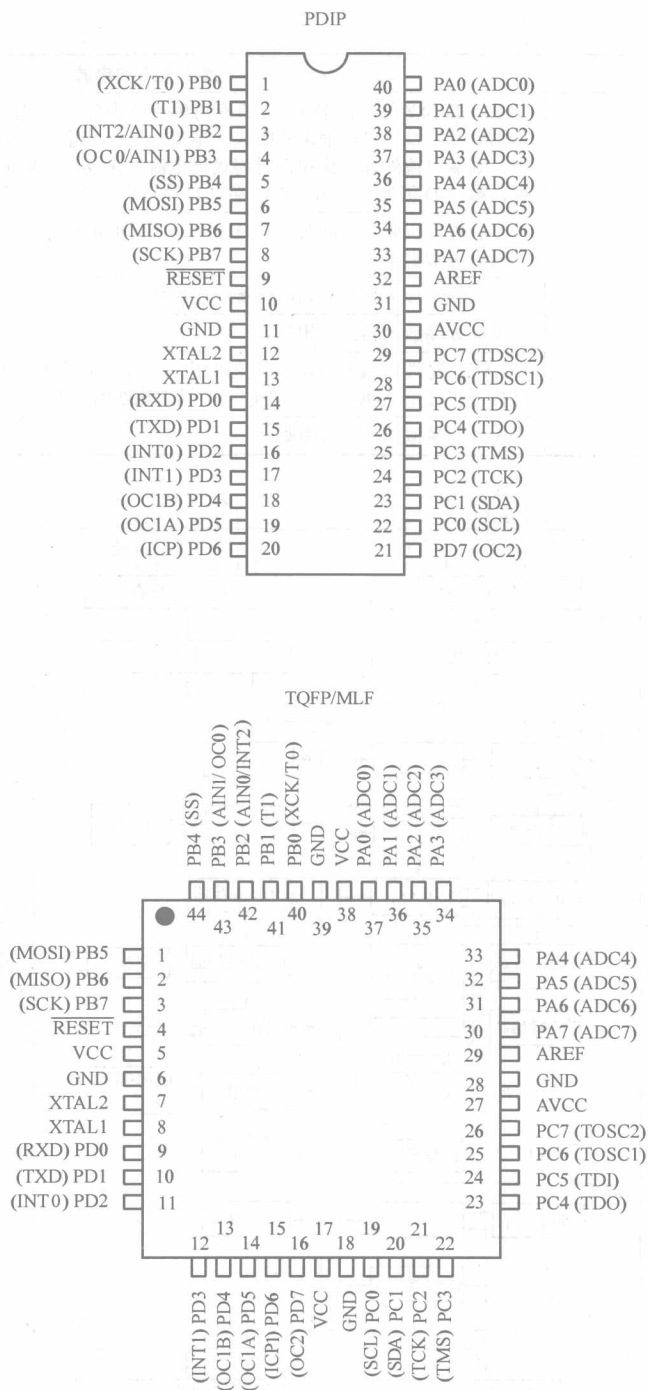


图 1-2 ATmega16/32 单片机的引脚

1.2 ATmega16/32 单片机 CPU 结构

ATmega16/32 单片机 CPU 的结构如图 1-3 所示。

ATmega16/32 单片机 CPU 采用了 Harvard 结构，即具有独立的程序和数据总线，而且

存储器也是独立的，程序存储器是可编程的 Flash 存储器。程序存储器里的指令通过一级流水线运行。当 CPU 在执行某一条指令时，将预先从程序存储器里读取下一条指令。这种结构使指令可以在每一个时钟周期内执行，从而显著提高了指令的执行速度。

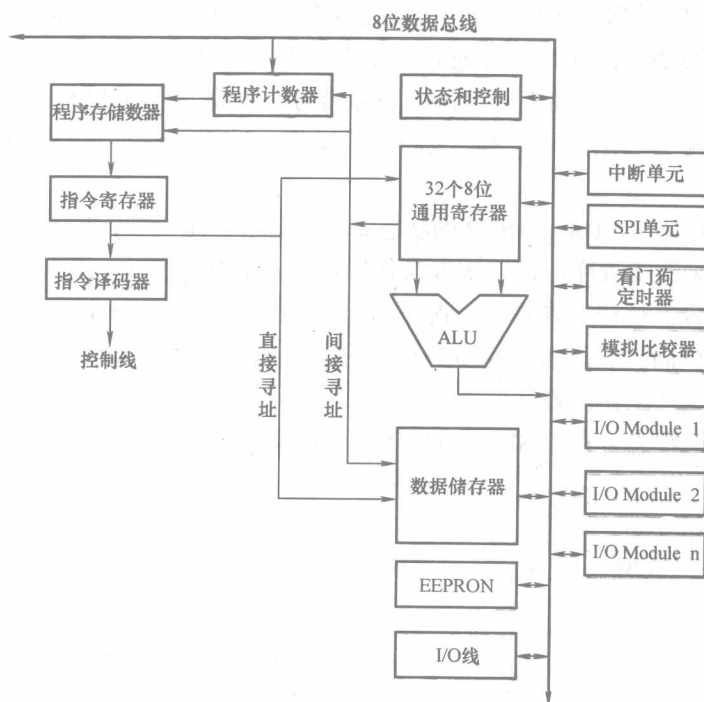


图 1-3 ATmega16/32 单片机 CPU 的结构

CPU 内的快速访问寄存器文件包括 32 个 8 位通用工作寄存器，访问时间为一个时钟周期。从而实现了单时钟周期的 ALU 操作。在典型的 ALU 操作中，两个位于寄存器文件中的操作数同时被访问，然后执行运算，结果再被送回到寄存器文件。整个过程只需一个时钟周期。寄存器文件里有 6 个寄存器，可以用作 3 个 16 位的间接寻址寄存器指针以寻址数据空间，实现高效的地址运算。其中一个指针还可以作为程序存储器查询表的地址指针，这些附加的功能寄存器即为 16 位的 X, Y, Z 寄存器。

ALU 支持寄存器之间以及寄存器和常数之间的算术和逻辑运算，也可以执行单寄存器操作。运算完成之后状态寄存器的内容得到更新以反映操作结果。

程序流程通过有/无条件的跳转指令和调用指令来控制，从而直接寻址整个地址空间。大多数指令长度为 16 位，即每个程序存储器地址都包含一条 16 位或 32 位的指令。

程序存储器空间分为两个区，即引导程序区（Boot 区）和应用程序区。这两个区都有专门的锁定位以实现读和读/写保护。用于写应用程序区的 SPM 指令必须位于引导程序区。

在中断和调用子程序时，返回地址的程序计数器（PC）保存于堆栈之中。堆栈位于通用数据 SRAM 中，因此其深度仅受限于 SRAM 的大小。在复位后，由于 SP 为 0000H，所有的程序必须首先初始化堆栈指针 SP。这个指针位于 I/O 空间，可以进行读写访问。数据 SRAM 可以通过 5 种不同的寻址模式进行访问。

AVR 存储器空间为线性的平面结构。AVR 有一个非常灵活的中断模块。控制寄存器位

于 I/O 空间。状态寄存器里有全局中断使能位。每个中断在中断向量表里都有独立的中断向量。各个中断的优先级与其在中断向量表的位置有关，中断向量地址越低，优先级越高。

I/O 存储器空间包含 64 个可以直接寻址的地址，作为 CPU 外设的控制寄存器、SPI，以及其他 I/O 功能，映射到数据空间即为寄存器文件之后的地址 0x20~0x5F。

(1) ALU 运算逻辑单元。ATmega16/32 的高性能 ALU 算术逻辑单元与 32 个通用工作寄存器直接相连。寄存器与寄存器之间、寄存器与立即数之间的 ALU 运算只需要一个时钟周期。ALU 操作分为三类，即算术、逻辑和位操作，此外还提供了支持有/无符号数和分数乘法的乘法器。

(2) 状态寄存器。ATmega16/32 的状态寄存器包含了最近执行的算术指令的结果信息。这些信息可以用来改变程序流程以实现条件操作。如指令集所述，所有 ALU 运算都将影响状态寄存器的内容。这样，在许多情况下就不需要专门的比较指令了，从而使系统运行更快速，代码效率更高。在进入中断服务程序时，状态寄存器不会自动保存，中断返回时也不会自动恢复。这些工作需要软件来处理。

ATmega16/32 状态寄存器 SREG 定义如图 1-4 所示。

Bt	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

图 1-4 状态寄存器 SREG 定义

①Bit 7——I：全局中断使能。

I 置位时，使能全局中断。单独的中断使能由其他独立的控制寄存器控制。如果 I 清零，则不论单独中断标志置位与否，都不会产生中断。任意一个中断发生后，I 清零，而执行 RETI 指令后，I 恢复置位以使能中断。I 也可以通过 SEI 和 CLI 指令来置位和清零。

②Bit 6——T：位复制存储。

位复制指令 BLD 和 BST 利用 T 作为目的或源地址。BST 把寄存器的某一位复制到 T，而 BLD 把 T 复制到寄存器的某一位。

③Bit 5——H：半进位标志。

半进位标志 H 表示算术操作发生了半进位，此标志对于 BCD 运算非常有用。

④Bit 4——S：符号位， $S=N \oplus V$ 。

S 为负数标志 N 与 2 的补码溢出标志 V 的异或。

⑤Bit 3——V：2 的补码溢出标志。

支持 2 的补码运算。

⑥Bit 2——N：负数标志。

表明算术或逻辑操作结果为负。

⑦Bit 1——Z：零标志。

表明算术或逻辑操作结果为零。

⑧Bit 0——C：进位标志。

表明算术或逻辑操作发生了进位。

(3) 通用寄存器。

ATmega16/32CPU 的通用寄存器结构如图 1-5 所示。

通用寄存器	7	0	Addr	
	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	X 寄存器, 低字节
	R27		\$1B	X 寄存器, 低字节
	R28		\$1C	Y 寄存器, 低字节
	R29		\$1D	Y 寄存器, 低字节
	R30		\$1E	Z 寄存器, 低字节
	R31		\$1F	Z 寄存器, 低字节

图 1-5 ATmega16/32CPU 的通用寄存器

如图 1-5 所示, 每个寄存器被分配给一个数据存储器地址, 将其直接映射到数据空间的前 32 个地址。虽然寄存器文件的物理实现不是 SRAM, 这种内存组织方式在访问寄存器方面具有极大的灵活性, 因为 X, Y, Z 寄存器可以设置为指向任意寄存器的指针寄存器。

大多数操作寄存器文件的指令都可以直接访问所有的寄存器, 而且多数这样的指令的执行时间为单个时钟周期。

(4) X, Y, Z 寄存器。

R26~R31 除了用作通用寄存器外, 还可以作为数据空间间接寻址用的地址指针。在不同的寻址模式中, 这些地址寄存器可以实现固定偏移量, 自动加 1 和自动减 1 功能, 如图 1-6 所示。

X 寄存器	15	XH		XL	0
	7		0	7	0
	R27 (\$1B)			R26 (\$1A)	
Y 寄存器	15	YH		YL	0
	7		0	7	0
	R29 (\$1D)			R28 (\$1C)	
Z 寄存器	15	ZH		ZL	0
	7		0	7	0
	R31 (\$1F)			R30 (\$1E)	

图 1-6 X, Y, Z 寄存器

(5) 堆栈指针。

堆栈指针主要用来保存临时数据、局部变量和中断/子程序的返回地址。堆栈指针总是

指向堆栈的顶部。要注意 AVR 的堆栈是向下生长的，即当新数据压入堆栈时，堆栈指针的数值将减小。堆栈指针指向数据 SRAM 堆栈区，在此聚集了子程序堆栈和中断堆栈。调用子程序和使能中断之前必须定义堆栈空间，且堆栈指针必须指向高于 0x60 的地址空间。使用 PUSH 指令将数据推入堆栈时指针减 1；而子程序或中断返回地址推入堆栈时指针将减 2。使用 POP 指令将数据弹出堆栈时，堆栈指针加 1；而用 RET 或 RETI 指令从子程序或中断返回时堆栈指针加 2。

AVR 的堆栈指针由 I/O 空间中的两个 8 位寄存器实现。实际使用的位数与具体器件有关，如图 1-7 所示。

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

图 1-7 SP 寄存器

1.3 ATmega16/32 单片机存储器

AVR 结构具有两个主要的存储器空间，即数据存储器空间和程序存储器空间。此外，还有 EEPROM 存储器以保存数据。这三个存储器空间都为线性的平面结构。

1. 可编程的 Flash 程序存储器

ATmega16 具有 16KB 的在线编程 Flash，用于存放程序指令代码，而 ATmega32 具有 32KB。因为所有的 AVR 指令为 16 位或 32 位，故而将 Flash 组织成 8KB×16 位或 16KB×16 位的形式。用户程序的安全性要根据 Flash 程序存储器的两个区，即引导（Boot）程序区和应用程序区分开来考虑。

Flash 存储器至少可以擦写 10000 次。ATmega16 的程序计数器（PC）为 13 位，因此可以寻址 8KB 的程序存储器空间，而 ATmega32 的程序计数器（PC）为 14 位，因此可以寻址 16KB 的程序存储器空间，常数可以保存于整个程序存储器地址空间。

程序存储器映像如图 1-8 所示，取指与执行时序如图 1-9 所示。

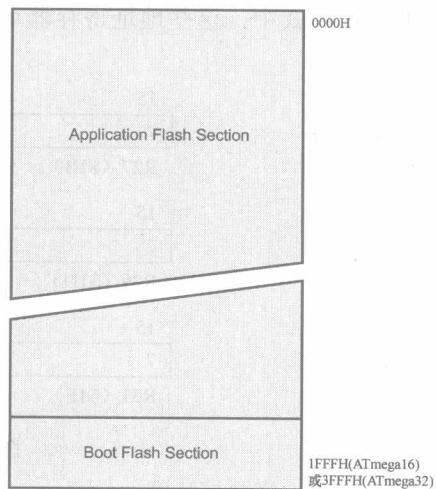


图 1-8 程序存储器映像

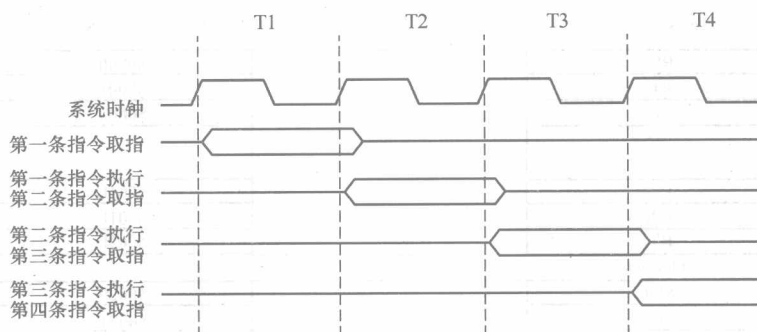


图 1-9 取指与执行时序

2. SRAM 数据存储器

ATmega16 和 ATmega32 的数据存储器组织分别如图 1-10 和图 1-11 所示。

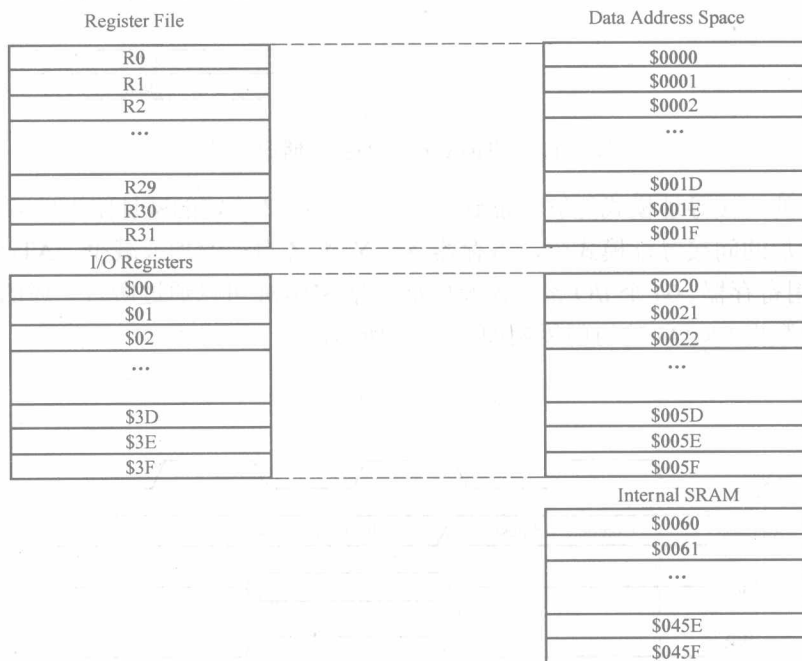


图 1-10 ATmega16 的数据存储器组织

前 1120 (ATmega16) 或 2144 (ATmega32) 个数据存储器包括了寄存器文件、I/O 存储器, 及内部数据 SRAM。起始的 96 个地址为寄存器文件与 64 个 I/O 存储器, 接着是 1024B (ATmega16) 或 2048B (ATmega32) 的内部数据 SRAM。

数据存储器的寻址方式分为 5 种, 即直接寻址、带偏移量的间接寻址、间接寻址、带预减量的间接寻址和带后增量的间接寻址。寄存器文件中的寄存器 R26~R31 为间接寻址的指针寄存器, 直接寻址范围可达整个数据区。

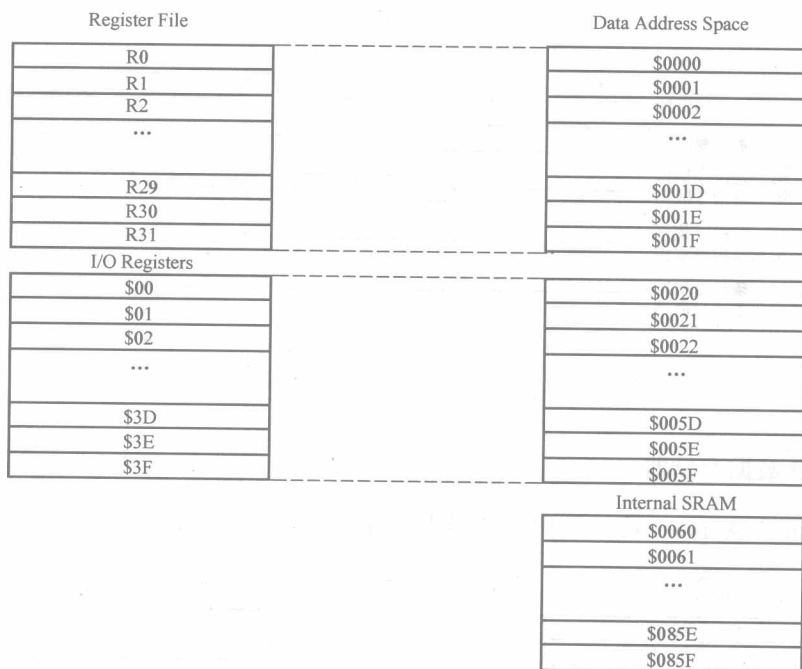


图 1-11 ATmega32 的数据存储器组织

带偏移量的间接寻址模式能够寻址到由寄存器 Y 和 Z 给定的基址附近的 63 个地址。在自动预减和后加的间接寻址模式中, 寄存器 X, Y 和 Z 自动增加或减少。ATmega16/32 的全部 32 个通用寄存器、64 个 I/O 寄存器及内部数据 SRAM 可以通过所有上述的寻址模式进行访问。片内数据 SRAM 的访问周期如图 1-12 所示。

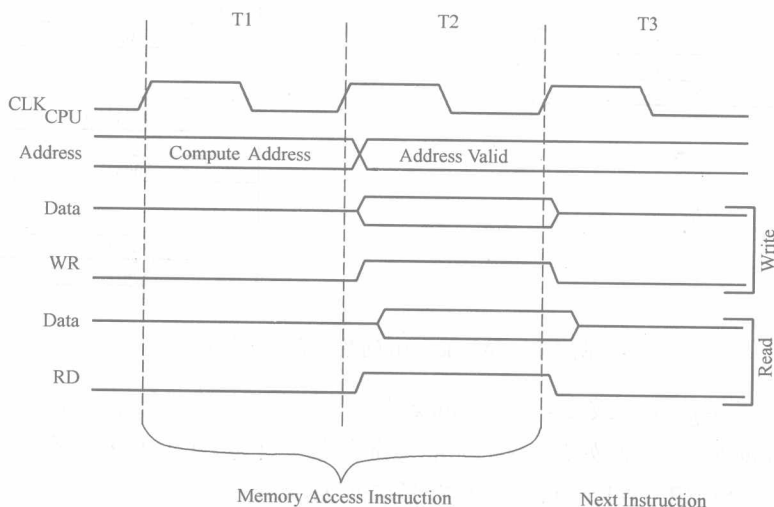


图 1-12 片内数据存储器访问周期

3. EEPROM 数据存储器

ATmega16 和 ATmega32 分别包含 512B 和 1024B 的 EEPROM 数据存储器。它是作为一

个独立的数据空间而存在的,可以按字节读写。EEPROM 的寿命至少为 100000 次擦除周期。EEPROM 的访问由地址寄存器、数据寄存器和控制寄存器决定。

1.4 ATmega16/32 单片机复位系统

ATmega16/32 是低电平复位,这点与目前常用的 MCS-51 系列单片机有所不同。

ATmega16/32 复位时所有的 I/O 寄存器都被设置为初始值,程序从复位向量处开始执行。复位向量处的指令必须是绝对跳转 JMP 指令,以使程序跳转到复位处理例程。如果程序永远不利用中断功能,中断向量可以由一般的程序代码所覆盖。复位逻辑电路图和复位电路的电气参数请参考相关资料。

复位源有效时,I/O 端口立即复位为初始值。此时不要求任何时钟处于正常运行状态。所有的复位信号消失之后,芯片内部的一个延迟计数器被激活,将内部复位的时间延长。这种处理方式使得在 MCU 正常工作之前,有一定的时间让电源达到稳定的电平。延迟计数器的溢出时间通过熔丝位 SUT 与 CKSEL 设定。

ATmega16/32 有 5 个复位源,分别是:

- (1) 上电复位。电源电压低于上电复位门限 VPOT 时,MCU 复位。
- (2) 外部复位。引脚 RESET 上的低电平持续时间大于最小脉冲宽度时 MCU 复位。
- (3) 看门狗复位。看门狗使能并且看门狗定时器溢出时复位发生。
- (4) 掉电检测复位。掉电检测复位功能使能,且电源电压低于掉电检测复位门限 VBOT 时 MCU 即复位。
- (5) JTAG AVR 复位。复位寄存器为 1 时 MCU 复位。

有关复位的具体介绍,请参考 ATmega16/32 的数据手册。

1.5 ATmega16/32 单片机中断

系统在正常运行主程序时,如果突然有一个重要的任务要马上处理,那么系统就要保存现在的工作(保护现场),然后去处理这个重要的任务,当任务执行完毕后恢复现场,再返回到原来的主程序继续运行,这就是中断的一般流程。

ATmega16/32 有二十多个不同的中断源,每个中断和复位在程序空间都有独立的中断向量。所有的中断事件都有自己的使能位。当使能位置位,且状态寄存器的全局中断使能位 I 也置位时,中断可以发生。程序存储区的最低地址默认为复位向量和中断向量。完整的向量列表请参见表 1-2。列表同时也决定了不同中断的优先级。向量所在的地址越低,优先级越高。RESET 具有最高的优先级,第二个为 INTO,即外部中断请求 0。通过置位通用中断控制寄存器(GICR)的 IVSEL,中断向量可以移至引导 Flash 的起始处,任一中断发生时,全局中断使能位 I 被清零,从而禁止了所有其他的中断。用户软件可以在中断程序里置位 I 来实现中断嵌套。此时所有的中断都可以中断当前的中断服务程序。执行 RETI 指令后,I 自动置位。