

AVR 单片机应用技术

李 勋 耿德根 编著



A1005671

北京航空航天大学出版社

<http://www.buaapress.com.cn>

内容简介

AVR 单片机是目前最新的单片机系列之一,它以高速度、大驱动能力和丰富的硬件资源而著称于微控制器系列。本书前两章系统地讲解了 AVR 单片机系列的硬件组成、运作原理及其指令集。在此基础上,第三、四两章进一步例示了 AVR 汇编语言程序设计技巧及片内各部件的用法。最后两章告诉读者如何构成单片机应用系统及进行软硬件开发。

本书取材先进,叙述条理清晰,语言通畅,例题丰富,可读性强。以高等院校有关专业师生及从事微机控制方面工作的广大科技人员为读者对象。

图书在版编目(CIP)数据

AVR 单片机应用技术/李勋等编著. —北京:北京航空航天大学出版社,2002.6

ISBN 7-81077-177-9

I. A… II. 李… III. 单片微型计算机, AVR
IV. TP368.1

中国版本图书馆 CIP 数据核字(2002)第 030287 号

AVR 单片机应用技术

李 勋 耿德根 编著

责任编辑 刘宝俊

责任校对 戚 爽

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn>

E-mail: pressell@publica.bj.cninfo.net

北京市宏文印刷厂印装 各地书店经销

*

开本:787×1092 1/16 印张:15.5 字数:397 千字

2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-177-9/TP·099 定价:23.00 元

前 言

AVR 单片机是目前最新单片机系列之一,其突出的特点在于速度高、片内硬件资源丰富、可作为真正意义上的单片机使用。

AVR 采用了现代微处理器流水管线预取指令技术,淘汰了机器周期的概念,以时钟周期为指令执行的最基本时间单位。通常时钟频率用 4 MHz~8 MHz,故最短指令执行时间为 250~125 ns。

这种单片机借鉴了某些机型的高速输入输出 HSIO 和可编程计数阵列 PCA 的概念,实现了本身的输入捕捉、输出比较和脉冲宽度调制输出功能。这是脉冲信号测量、开关量按时控制及某些直流马达调速的得力工具。AVR 除继承了前代器件的异步串行口外,尚增加了高速同步串行口 SPI,因此可方便灵活地与其他器件进行串行通讯。AVR 监视定时器具有独立的时钟,故在掉电方式下仍可正常运行。这样就增加了使 MCU 从掉电方式下苏醒过来的手段。该系列各器件大多内含模拟比较器,有的尚有内部 A/D 转换器,为处理模拟量输入提供了条件。

AVR 摒弃了 EPROM 和掩膜 ROM 方案,以擦写很方便的 Flash 作为程序存储器,且其容量档次齐全,用户可根据实际需要进行选择。加之片内 SRAM 和 EEPROM 可满足一般要求,故通常无系统扩展的必要。所有 I/O 线均可用作通用输入输出,因此可实现与外设之间的直接接口,联络信号由软件产生。

作者认为有必要把这种优秀的芯片介绍给我国广大的有关科技人员,使之成为他们在微控制领域的得力助手。本书的特点是在吃透原文文献的基础上,结合作者多年实践经验写成的。书内大量的例题中,多处体现着作者的思路。这也正是称本书为“编著”的原因之所在。

全书共六章,其中第六章由耿德根高级工程师供稿,其余五章均由李勋教授撰写。本书的两位作者希望看到广大读者对本书的意见和评价。

作 者

天津工业大学

2002.2

目 录

第一章 体系结构及运作原理

1.1 AT90 系列	(1)
1.2 内部结构	(1)
1.2.1 结构框图	(1)
1.2.2 通用寄存器组合与 ALU	(4)
1.2.3 存储器组织	(5)
1.2.4 程序和数据寻址方式	(6)
1.2.5 I/O 存储器	(9)
1.2.6 定时/计数器 0	(12)
1.2.7 定时/计数器 1	(15)
1.2.8 监视定时器	(23)
1.2.9 EEPROM 读写	(25)
1.2.10 串行涉外接口 SPI	(27)
1.2.11 通用异步接收发送器	(32)
1.2.12 模拟比较器	(38)
1.2.13 复位控制电路	(39)
1.2.14 中断系统	(41)
1.2.15 I/O 端口	(47)
1.3 封装与引脚功能	(57)
1.3.1 引脚功能	(57)
1.3.2 CPU 时钟	(59)
1.3.3 指令执行的时间概念	(59)
1.3.4 与片外数据存储器接口	(61)
1.4 MCU 的休眠方式	(62)
1.4.1 闲置方式	(62)
1.4.2 掉电方式	(63)
1.5 片内存储器的烧录	(63)
1.5.1 并行烧录	(63)
1.5.2 串行下载	(70)

第二章 指令系统

2.1 数据传送指令	(74)
2.1.1 工作寄存器间的数据传送指令	(74)
2.1.2 工作寄存器的数据装载指令	(74)
2.1.3 工作寄存器的数据存储指令	(77)
2.1.4 程序存储器的读指令 LPM	(78)
2.1.5 输入/输出指令	(78)

2.1.6	堆栈操作指令	(79)
2.2	算术和逻辑运算指令	(81)
2.2.1	加法类指令	(81)
2.2.2	减法类指令	(82)
2.2.3	逻辑与指令	(85)
2.2.4	逻辑或指令	(86)
2.2.5	逻辑异或指令	(87)
2.3	分支指令	(88)
2.3.1	无条件跳转指令	(89)
2.3.2	子程序调用指令	(89)
2.3.3	返回指令	(90)
2.3.4	条件跨越指令	(90)
2.3.5	基本条件分支指令	(92)
2.3.6	派生条件分支指令	(92)
2.4	位操作及其他指令	(98)
2.4.1	I/O 寄存器位置位和位清 0 指令	(98)
2.4.2	状态寄存器位置位和位清 0 指令	(99)
2.4.3	位传送指令	(100)
2.4.4	移位指令	(101)
2.4.5	其他指令	(103)
第三章 汇编语言程序设计示范		
3.1	AVR 程序的书写规则	(104)
3.1.1	汇编指令	(104)
3.1.2	表达式	(110)
3.2	BCD 运算	(114)
3.2.1	数制转换	(114)
3.2.2	BCD 加减运算	(118)
3.2.3	BCD 运算测试程序	(119)
3.3	多倍精度运算	(120)
3.3.1	多字节加减运算	(120)
3.3.2	16 位运算	(121)
3.4	数据块传送	(123)
3.4.1	程序存储器至数据存储器的数据传送	(123)
3.4.2	数据存储器两区域间的数据块传送	(123)
3.4.3	数据块传送测试程序	(124)
3.5	乘除法运算	(124)
3.5.1	8 位无符号数乘法	(124)
3.5.2	8 位带符号数乘法	(126)
3.5.3	8 位无符号数除法	(127)

3.5.4	8 位带符号数除法	(128)
3.6	数据排序	(129)
3.6.1	冒泡排序算法	(129)
3.6.2	数据排序测试程序	(130)
3.7	数字滤波	(131)
3.7.1	去极值平均滤波法	(131)
3.7.2	8 点移动平均值滤波法	(132)
第四章 片内硬件资源的应用		
4.1	对片内 EEPROM 的访问	(135)
4.1.1	对片内 EEPROM 的随机读写	(135)
4.1.2	对片内 EEPROM 的顺序读写	(136)
4.2	端口的通用 I/O 操作	(139)
4.3	监视定时器的应用	(142)
4.3.1	启用和中期管理	(142)
4.3.2	顺序控制	(144)
4.4	定时/计数器 0 的应用	(147)
4.4.1	定时功能的实现	(147)
4.4.2	计数功能的实现	(149)
4.5	定时/计数器 1 的应用	(151)
4.5.1	定时功能应用实例	(151)
4.5.2	输出比较功能的应用	(153)
4.5.3	输入捕捉功能的应用	(156)
4.5.4	频率测量	(159)
4.5.5	脉宽调制	(160)
4.6	模拟比较器的应用	(162)
4.6.1	模拟比较器的工作设定	(162)
4.6.2	简易 A/D 转换器	(164)
4.6.3	逐步逼近式 A/D 转换	(166)
4.7	串行口的应用	(168)
4.7.1	同步串行传送	(168)
4.7.2	异步串行传送	(170)
4.7.3	软件 UART	(173)
4.8	休眠方式的应用	(178)
4.8.1	掉电方式的进入和退出	(178)
4.8.2	闲置方式的进入和退出	(180)
第五章 接口技术		
5.1	与显示器接口	(181)
5.1.1	发光二极管显示器	(181)
5.1.2	液晶显示器	(185)

5.2 与键盘接口	(195)
5.2.1 独立按键接口	(195)
5.2.2 矩阵式键盘接口	(196)
5.2.3 键盘和显示器复合接口	(200)
5.3 与BCD码拨盘接口	(212)
5.3.1 与端口直接接口	(212)
5.3.2 通过门电路接口	(213)
5.4 与微型打印机接口	(215)
5.4.1 TP μ P微型打印机	(215)
5.4.2 AVR与TP μ P—40A接口	(216)
5.5 片内A/D转换器	(217)
5.5.1 内部结构及运作原理	(217)
5.5.2 ADC应用程序	(222)
第六章 开发工具及应用	
6.1 AVR实时在线仿真器ICE200	(226)
6.2 JTAG ICE仿真器	(226)
6.3 开发下载实验器SL—AVR+	(227)
6.4 AVR集成开发环境(IDE)	(232)
6.4.1 AVR Assembler汇编器	(232)
6.4.2 AVR Studio	(235)
6.4.3 AVR Prog	(236)

第一章 体系结构及运作原理

AVR 单片机包括 AT90、ATtiny 和 ATmega 三个系列。其体系结构大同小异,指令系统相互包容。本章将以 AT90 系列为主线介绍 AVR 微控制器的机理。

1.1 AT90 系列

AT90 系列包括 AT90S1200、AT90S2313、AT90S/LS2323 和 AT90S/LS2343、AT90S/LS2333 和 AT90S/LS4433、AT90S4414/8515、AT90S/LS4434 和 AT90S/LS8535 以及 AT90C8534 等众多品牌的微控制器。它们都是 8 位 CMOS 低功耗器件,其基本框架为 AVR RISC 体系结构,即被命名为 AVR 的精减指令集计算机体系结构。

AT90 系列各单片机内含容量不等的闪速程序存储器(Flash Memory,简称 Flash),可反复擦写至少 1 000 次。这极大地方便了产品开发和软件修改。这种存储器中的程序可由 PC 机串行下载,亦可在通用写入器上以并行方式写入。

AVR 单片机最大的特点是低功耗和高速度。掉电方式、闲置方式至工作方式下的耗电约在 $1\ \mu\text{A}\sim 2.5\ \text{mA}$ 之间。此类单片机每个时钟周期可执行一条指令。在 12 MHz 频率下,指令的吞吐量为 12 MIPS,这是一般 MCS—51 单片机速度的 12 倍。

AVR 片内含有模拟比较器,与输入捕捉功能配合,可进行多种模拟控制和转换。

AVR 单片机含有内部上电复位电路,因此其低电平有效的 RESET 引脚可直接连到 V_{CC} 上,省却片外复位电路的安排。AT90S1200 在片内还有 1 MHz 的 RC 振荡器,这样就可作为无外加元件的真正意义上的单片机用了。

表 1.1 中给出了 AT90 系列各微控制器片内资源的简要情况。

1.2 内部结构

AVR 微控制器大家族体系结构相同,各成员间内部结构稍有差异。本节将以 AT90S4414/8515 为代表,详细讲述 AVR 单片机内部各部件的组织结构及其运作原理。

1.2.1 结构框图

大家知道,计算机的核心为运算器和控制器,两者合称为中央处理单元,即 CPU。所谓单片机,是把 CPU、一定容量的存储器以及若干输入/输出接口等部件集成在一块小小硅片上的微型计算机,简称单片机。因主要用于控制功能,故亦称微控制器,有时写作 MCU。

如图 1.1 所示,作为 AVR 单片机内核的 CPU 包括下列各部件:

- 运算器——算术逻辑单元 ALU 以及与其配合工作的通用寄存器组 and 状态寄存器;
- 控制器——程序计数器 PC、指令寄存器、指令译码器和定时与控制部件。

AVR 单片机的存储器有用来存放程序的闪速存储器 Flash、存放数据和中间结果的静态

表 1.1 AT90 系列和 ATtiny 系列微控制器简况

型 号	闪存 /KB	EEPROM /B	SRAM /B	指令 条数	I/O 引脚	引脚 总数	外部 中断	SPI	UART	8 位 定时器	16 位 定时器	PWM	监视 定时器	10 位 AD 通道	模拟 比较器	V _{CC} /V	时钟频率 /MHz
AT90S1200	1	64	—	89	15	20	1	√	—	1	—	—	√	—	√	2.7~6 4~6	0~4 0~12
AT90S2313	2	128	128	118	15	20	2	√	√	1	1	√	√	—	√	2.7~6 4~6	0~4 0~10
AT90S2323	2	128	128	118	3	8	1	√	—	1	—	—	√	—	—	4~6	0~10
AT90LS2323	2	128	128	118	3	8	1	√	—	1	—	—	√	—	—	2.7~6	0~4
AT90S2343	2	128	128	118	5	8	1	√	—	1	—	—	√	—	—	4~6	0~10
AT90LS2343	2	128	128	118	5	8	1	√	—	1	—	—	√	—	—	2.7~6	0~4
AT90S2333	2	128	128	118	20	28/32	2	√	√	1	1	√	√	—	√	4~6	0~8
AT90LS2333	2	128	128	118	20	28/32	2	√	√	1	1	√	√	—	√	2.7~6	0~4
AT90S4433	4	256	128	118	20	28/32	2	√	√	1	1	√	√	—	√	4~6	0~8
AT90LS4433	4	256	128	118	20	28/32	2	√	√	1	1	√	√	—	√	2.7~6	0~4
AT90S4414	4	256	256	118	32	40/44	2	√	√	1	1	√	√	—	√	2.7~6 4~6	0~4 0~8
AT90S8515	8	512	512	118	32	40/44	2	√	√	1	1	√	√	—	√	2.7~6 4~6	0~4 0~8
AT90S4434	4	256	256	118	32	40/44	2	√	√	2	1	√	√	8	√	4~6	0~8
AT90LS4434	4	256	256	118	32	40/44	2	√	√	2	1	√	√	8	√	2.7~6	0~4
AT90S8535	8	512	512	118	32	40/44	2	√	√	2	1	√	√	8	√	4~6	0~8
AT90LS8535	8	512	512	118	32	40/44	2	√	√	2	1	√	√	8	√	2.7~6	0~4
AT90C8534	8	512	256	118	7	48	2	—	—	1	1	—	—	8	—	3.3~6	0~1.5
ATtiny10	1	—	—	90	—	8	1	—	—	1	—	—	√	—	√	2.7~5.5 4.0~5.5	0~2 0~6
ATtiny11	1	—	—	90	—	8	1	—	—	1	—	—	√	—	√	2.7~5.5 4.0~5.5	0~2 0~6
ATtiny12	1	64	—	90	—	8	1	√	—	1	—	—	√	—	√	1.8~5.5 2.7~5.5 4~5.5	0~1 0~4 0~8

随机存取存储器 SRAM 以及存放某些常数的电可擦除的可编程只读存储器 EEPROM。

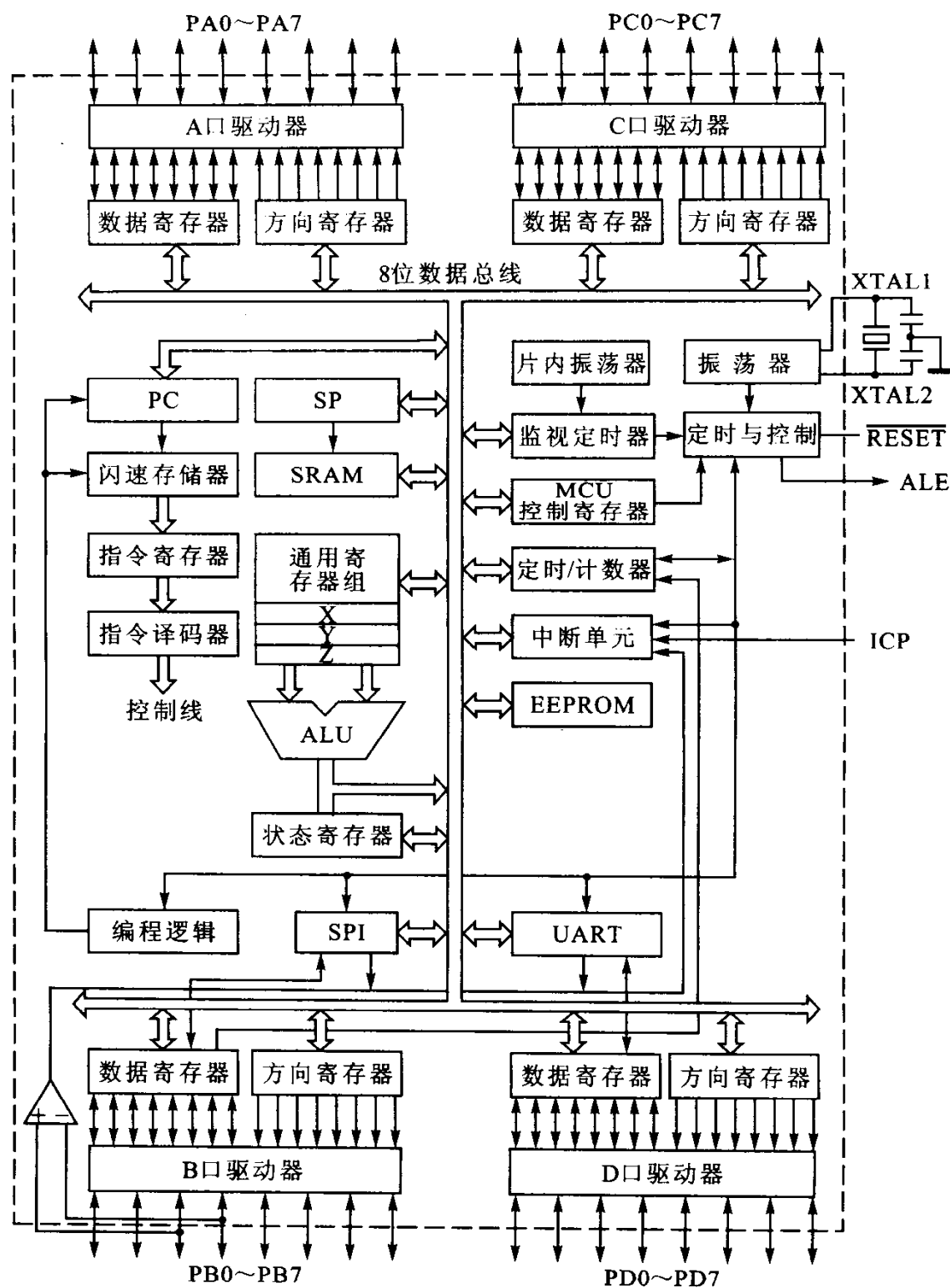


图 1.1 AVR 单片机结构框图

AT90S4414 和 AT90S8515 两机型只是其片内存储器容量不同,其他方面完全一致。前者片内含 4 KB 闪速存储器、256 字节 EEPROM 和 256 字节 SRAM;后者的闪速存储器为 8 KB,其 EEPROM 和 SRAM 均为 512 字节。

CPU 的主要功能是周而复始地读取指令并执行指令。首先由程序计数器 PC 提供行将执行的指令在程序存储器中的地址;该指令代码由那里被读取出来送指令寄存器暂存,进而由指令译码器对此代码进行译码;控制器根据译码结果向运算器各有关部件发出控制信号,最后运算器在这些信号的控制下完成该指令所要求的操作。

单片机与外部世界的沟通靠输入/输出(I/O)接口部件进行。AT90S4414/8515 单片机各

有 A、B、C、D 等四个并行 I/O 端口,辖 32 条 I/O 线。在图 1.1 中这些 I/O 线被标作 PA0~PA7 和 PB0~PB7 等。每个并行端口由驱动器、数据寄存器和数据方向寄存器组成。与外界的串行通讯由异步串行口 UART 和同步串行口 SPI 负责。其他涉外部件尚有定时/计数器和中断单元等。

1.2.2 通用寄存器组合与 ALU

如图 1.2 所示,AVR CPU 含 32 个 8 位通用工作寄存器 R0~R31,它们作为一个整体被称作通用寄存器组合。但其前 16 个寄存器(R0~R15)和后 16 个(R16~R31)之间,有共同点,也有差异。

按 AVR 指令系统的规定,所有涉及到寄存器操作的指令都可直接访问这 32 个工作寄存器。所谓直接访问,是指在 16 位指令代码中,除 11 位操作码用来限定所要求的操作外,尚以 5 位码直接给出该指令所涉及到的寄存器编号 0~31。或者在 16 位代码中有 6 位操作码的情况下,各以 5 位码分别指定目的寄存器和源寄存器编号。以上单寄存器和双寄存器的操作均可在一个时钟周期内完成。

然而,寄存器与立即数间的操作指令,却只能涉及到后 16 个寄存器,即 R16~R31。

从图 1.2 中可以看到,每个工作寄存器除有编号外,尚分配有一个数据存储器地址。这样,这些寄存器就被直接映射到用户数据空间的最低 32 个地址单元中了。尽管从物理上讲它们并不是像 SRAM 各存储单元那样构成的,但这种存储组织却为访问寄存器提供了很大的灵活性,因为 X、Y 和 Z 三个寄存器可被设置成访问该组合中任一寄存器的地址指针。

组合中最高编号的 8 个寄存器,即 R24~R31,可作 8 位寄存器独立运作,亦可组成寄存器对,作 16 位寄存器使用。这四个寄存器对中,R24:R25(寄存器对的两个字节间以冒号分割)只作数据寄存器。而 R26:R27(合称 X 寄存器)、R28:R29(Y 寄存器)和 R30:R31(Z 寄存器)三个寄存器对在通用功能的基础上又增加了一些专门功能。它们可作对数据空间间接寻址时的地址指针,因此有时也称其为地址寄存器。在不同的寻址方式中,这些地址寄存器可存放基准地址,可自动增量,亦可自动减量。

图 1.3 把图 1.2 中最高编号的六个寄存器画成了 16 位寄存器形式。

应当指出,用户可将任意两个工作寄存器作为寄存器对使用,不过,它们不具备上述三个指针寄存器的功能,也不像 R24:R25 那样有专用指令支持。

如图 1.1 所示,AVR ALU 直接与这 32 个工作寄存器相连,等于有 32 个累加器。这就消除了一些 MCU 中大多数算术逻辑运算都要通过累加器进行的瓶颈现象。此组合中寄存器间的操作,在一个时钟周期内即可完成。ALU 操作有三个主要类型:算术运算、逻辑运算和位操作。

7	0
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寄存器高字节
R28	\$1C—Y寄存器低字节
R29	\$1D—Y寄存器高字节
R30	\$1E—Z寄存器低字节
R31	\$1F—Z寄存器高字节

图 1.2 AVR CPU 通用工作寄存器

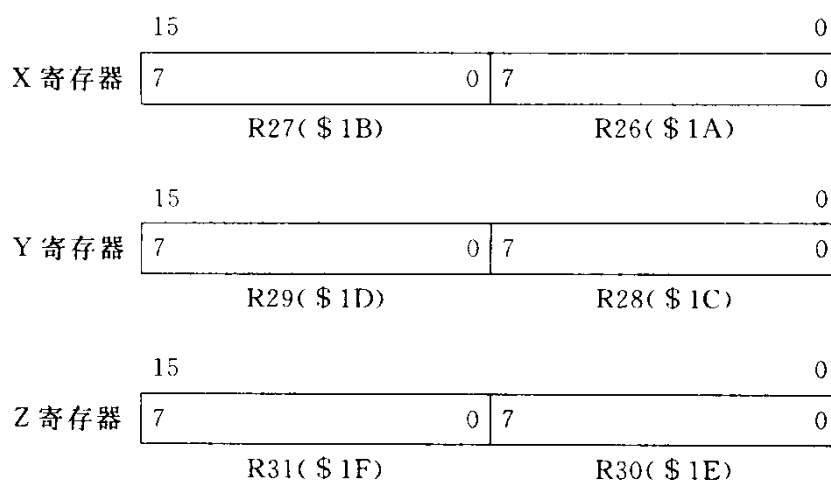


图 1.3 X、Y 和 Z 寄存器

1.2.3 存储器组织

AVR 存储器包括程序存储器 Flash、数据存储器 SRAM 和 EEPROM。

1. 可编程闪速存储器 Flash

AT90S4414 片内含 4 KB 系统内可编程的闪速存储器,简称 Flash,用来存放程序。AT90S8515 有 8 KB Flash。因为所有指令均为 16 位或 32 位字长,所以这里的 Flash 被组织成 $2\text{ K} \times 16$ 和 $4\text{ K} \times 16$ 形式。这种闪速存储器至少可以经受 1 000 次擦写操作。AT90S4414/8515 的程序计数器 PC 为 11/12 位宽,这样就可访问到全部 2 048/4 096 个地址单元。

闪速存储器的程序下载和对程序存储器的寻址方式将另节叙述。

2. 片内和片外 SRAM 数据存储器

图 1.4 所示为 AT90S4414/8515 静态随机存取存储器 SRAM 的组织形式。应当指出,AVR 程序存储器 Flash 和数据存储器 SRAM 分别占据两个独立的地址空间。AT90S4414/8515 Flash 的地址范围为 $\$000 \sim \$7\text{FF}/\$ \text{FFF}$;而 SRAM 所占空间则是 $\$0000 \sim \$ \text{FFFF}$ 。由图 1.4 可见,AVR 的片内和片外数据存储器 SRAM 是一个连续的地址空间,总容量可达 64 KB。最低 32 个地址单元 $\$0000 \sim \001F 系寄存器组合;其后 64 个单元 $\$0020 \sim \005F 为 I/O 寄存器;片内数据存储器 SRAM 占据此后的 256/512 个地址单元 $\$0060 \sim \$015\text{F}/\$025\text{F}$ 。这样,SRAM 空间的片内部分计有 352/608 个字节。

片外 SRAM 根据需要配置,可有可无,可大可小,其最低地址为 $\$0160/\0260 ,最高地址可达 $\$ \text{FFFF}$ 。指令本身并不区分片内和片外 SRAM,只要访问数据存储空间的地址超出片

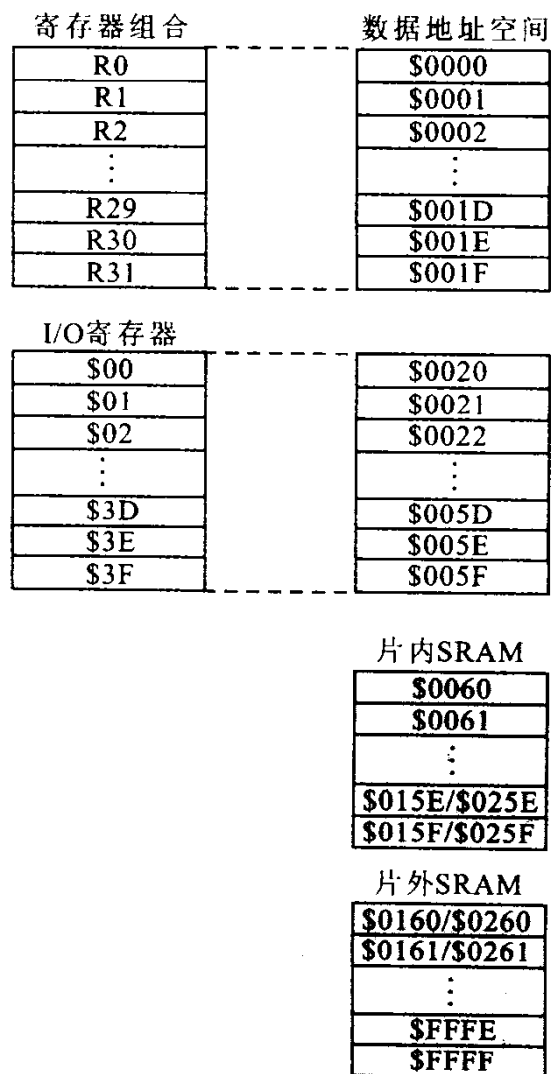


图 1.4 SRAM 组织

内数据 SRAM 的范围,那么这条指令就会自动去访问片外数据 SRAM。当访问片内数据空间时,读写选通信号引脚 $\overline{RD}$ 和 $\overline{WR}$ 在整个访问周期都处于无效状态。片外 SRAM 的操作,可开放也可禁止,这由 MCU 控制寄存器 MCUCR 中的 SRE 位来决定。

与对片内 SRAM 的访问操作相比,每访问片外 SRAM 一个字节,就要多用一个时钟周期。这就意味着,在计算对数据存储空间某字节的读写指令(例如 LD、ST、LDS、STS、PUSH 和 POP)执行时间时,若涉及到片外 SRAM,则勿忘增加一个时钟周期。如果堆栈设置在片外 SRAM 中的话,那么中断响应和返回、子程序调用和返回都要额外多花费两个时钟周期,因为要进行两个字节返回地址的入栈和出栈操作。当片外 SRAM 使用等待状态时,每个字节的操作要增加两个时钟周期。其结果是,数据传送指令要额外多用两个时钟周期,而中断响应和返回以及子程序调用和返回要比指令手册上给出的时间多出 4 个时钟周期。

对数据存储器有五种不同的寻址方式:直接寻址、带偏移量的间接寻址、间接寻址、先减量间接寻址和后增量间接寻址。在寄存器组合中,R26~R31 以可作间接寻址指针寄存器为其特色。

直接寻址方式可以访问整个数据空间。

偏移量间接寻址方式的访问范围可达基准地址后的 63 个地址单元,基准地址由 Y 或 Z 寄存器的内容给出。

当使用先自动减量或后自动增量的间接寻址方式时,地址寄存器 X、Y 或 Z 之内容将自动减量或增量。

AT90S4414/8515 的 32 个通用工作寄存器、64 个 I/O 寄存器、256/512 字节片内 SRAM 以及自由选配的片外 SRAM,均可通过这五种寻址方式访问到。

3. EEPROM 数据存储器

如前所述,AT90S4414/8515 内含 256/512 个字节 EEPROM 数据存储器。它占据独立的数据空间,对此空间可进行单一字节的读写操作。该 EEPROM 的寿命至少可耐 100 000 次擦写操作。有关 CPU 对 EEPROM 的读写、EEPROM 的地址寄存器、数据寄存器和控制寄存器等项内容,稍后将另题讲述。

1.2.4 程序和数据的寻址方式

AVR RISC 微控制器以强力高效的寻址方式支持对程序存储器 Flash 和数据存储器(包括 SRAM、寄存器组合及 I/O 存储器)的访问。寻址方式告诉 CPU 通过什么途径才能找到操作数或下一次行将执行的指令之地址。

1. 单寄存器直接寻址

由图 1.5 可见,在这种寻址方式下,指令的 16 位代码中,其高 11 位为操作码 OP,另 5 位 d 用来直接指定被寻址的寄存器编号。操作数便含在寄存器 d(即 Rd)中。

2. 双寄存器直接寻址

如图 1.6 所示,在双寄存器直接寻址方式下,指令代码中 5 位码 d 用来指定目的寄存器 Rd 的编号,另 5 位码 r 则给出寄存器组合中另一寄存器的编号。两操作数分别位于寄存器 r(即 Rr)和寄存器 d(Rd)中,操作结果将被置于 Rd 内。

应当指出,这些寻址方式图中,d 和 r 在代码中的位置不过是示意性的。在实际代码形成中,d 不一定占据最低 5 位,也有 5 位码不连续的情况,详见指令系统一章。

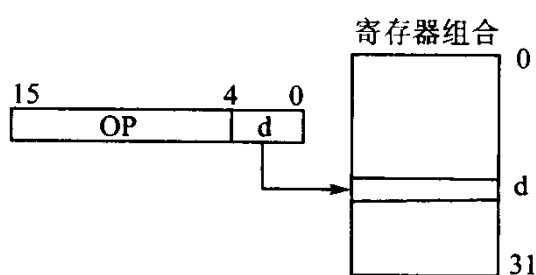


图 1.5 单寄存器直接寻址

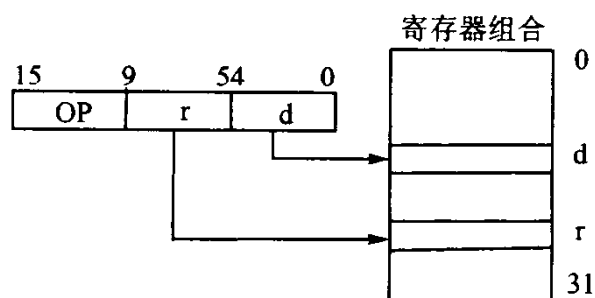


图 1.6 双寄存器直接寻址

3. I/O 直接寻址

这种寻址方式仅用于 IN(输入)和 OUT(输出)指令中。

在图 1.7 所示的这种寻址方式中,指令字的 6 位码 p 代表端口、定时器和配置寄存器等 I/O 寄存器地址。5 位码 n 用来指定寄存器组合中某寄存器为目的寄存器 R_d (IN 指令)或源寄存器 R_r (OUT 指令)。

4. 数据直接寻址

这是寄存器组合中某寄存器对数据空间某地址单元进行读写的一种寻址方式。如图 1.8 所示,在双字长的指令代码中,5 位码用来指定某寄存器作目的寄存器 R_d (存放由数据空间读入的数据),或作源寄存器 R_r (把其中数据写入数据空间)。指令代码的低 16 位(16 LSBs)直接给出数据空间某单元地址,数据即从这里读出,或写入此处。

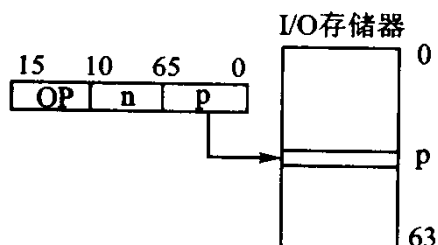


图 1.7 I/O 直接寻址

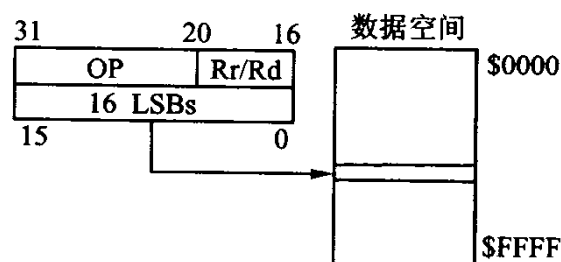


图 1.8 数据直接寻址

5. 带偏移量的数据间接寻址

图 1.9 所示为一种数据间接寻址的实现过程。这里被访问的数据字节之地址不便直接给出,而以基准地址加偏移量的形式间接指定出来。在这种寻址方式下,16 位指令代码中,6 位码 a 作为一个取值范围等于 $0 \sim 63$ 的偏移量,而 Y 或 Z 寄存器之 16 位内容为基准地址,两者之和(Y 或 $Z + a$)便是操作数在数据空间中的地址。指令码中的 n 照例仍负责指定寄存器组合中的某寄存器为目的寄存器 R_d (LDD 指令)或源寄存器 R_r (STD 指令)。

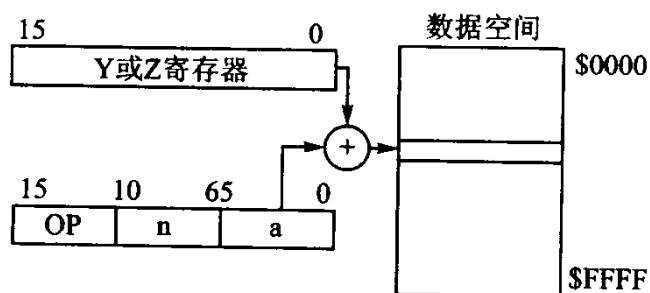


图 1.9 带偏移量的数据间接寻址

6. 数据间接寻址

这是另一种访问数据空间的寻址方式。所谓访问数据空间,就是在寄存器组合和数据空间之间进行数据传送。把数据从数据空间传送到寄存器组合叫作装载(LD,亦称读入);反向传送为存储(ST,写出)。

在数据间接寻址方式下,如图 1.10 所示,X、Y 和 Z 寄存器均可作地址指针寄存器。就是说,它们的 16 位内容便是操作数在存储空间中的地址。

7. 先减量数据间接寻址

有时需要对数据空间中的一个数据块或一张表格进行从其末地址开始的逐项访问,这种寻址方式将为之提供极大的方便。

在这种方式下,如图 1.11 所示,X、Y 或 Z 寄存器的内容在操作前先自动减 1,然后以其减量后的新内容作为操作数在数据空间中的地址。数据或由某源寄存器 R_r 写至此地址单元,或从这里读入到某目的寄存器 R_d 中。

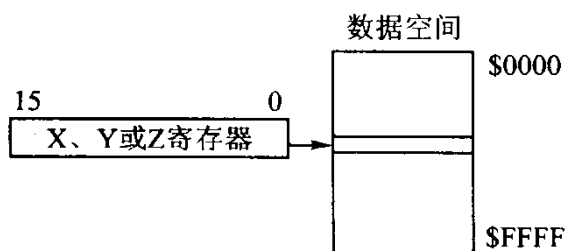


图 1.10 数据间接寻址

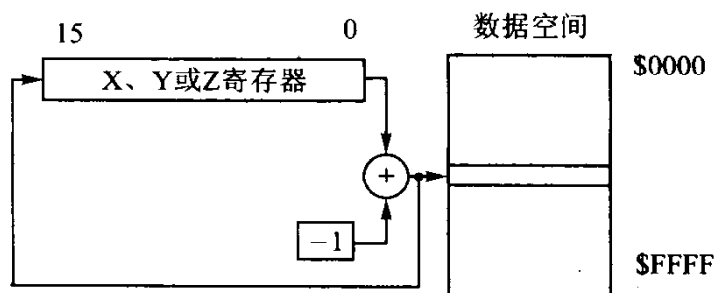


图 1.11 先减量数据间接寻址

当然,在数据间接寻址方式下,不管是偏移量间址、先减量间址、抑或后增量间址,都要在指令代码中以 5 位码指定 R_d 或 R_r。

8. 后增量数据间接寻址

后增量间接寻址方式便于对数据空间中的表格或数据块进行自首地址开始的读写操作。就其寻址过程讲,它和间接寻址并无不同,即以 X、Y 或 Z 等三个地址指针寄存器的内容为操作数的地址。只不过间接寻址后,指针内容保持不变,而后增量间址操作后,指针内容自动增 1,为下次访问做好准备。图 1.12 清楚地演示了这个过程。

9. 程序存储器常数寻址

这是 LPM 指令自程序存储器中读取常数所用的一种寻址方式。前已提及,AVR 单片机的程序存储器每个地址单元存放一个 16 位字。在程序存储器常数寻址方式下,地址指针寄存器 Z 的高 15 位用来指定程序存储器中的某个地址单元(见图 1.13),而 Z 的最低位将对该单元的两个字节做出选择,0 选中低字节,1 选中高字节。

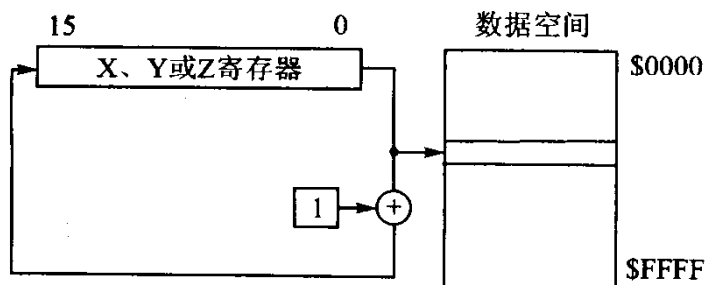


图 1.12 后增量数据间接寻址

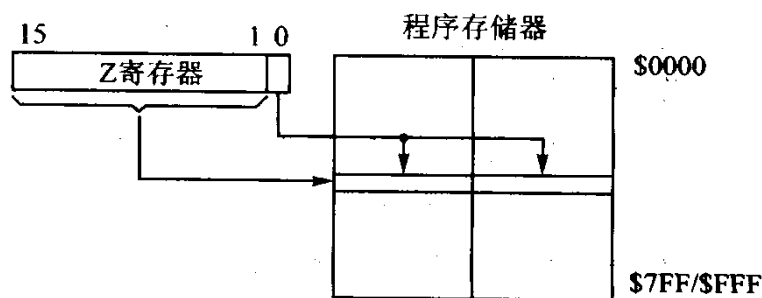


图 1.13 程序存储器常数寻址

10. 程序间接寻址

这是分支指令(跳转 IJMP 和子程序调用 ICALL)所用的一种寻址方式。所谓分支,即从某个非顺序地址单元继续执行程序。由图 1.14 可见,程序间接寻址方式告诉 CPU,地址指针寄存器 Z 的内容代表程序存储空间中的一个地址,接下来应去执行此地址单元所存放的指

令。因为程序计数器 PC 的使命是提供下一次要执行的指令的地址,所以这种寻址方式的运作实际上相当于把 Z 的内容送到 PC 中去。

11. 程序相对寻址

数据寻址是 CPU 寻址操作数的过程;而程序寻址(程序间接寻址、程序相对寻址、有些 AVR 品牌尚用程序直接寻址)则是寻找下一次行将执行的指令代码的过程。在跳转(JMP)和子程序调用(CALL)指令码中,以若干位码的形式直接给出跳转的目的地址或子程序的入口地址,这就是程序直接寻址方式。程序间接寻址则以指针寄存器 Z 的内容作为下一次要执行的指令的地址。

图 1.15 示意了程序相对寻址的运作过程。在执行相对跳转 RJMP 和相对调用 RCALL 指令时,目的地址是以基准地址加偏移量的形式给出的,即 $PC + k + 1$ 。这里 PC 是本 RJMP 或 RCALL 指令代码所在的地址,PC+1 则为其后一指令的地址,偏移量 k 便由此开始算起。例如,若目的地址为 $PC + 2$,则 $k = PC + 2 - (PC + 1) = 1$;若目的地址是 $PC - 2$,则 $k = PC - 2 - (PC + 1) = -3$,如此等等。不难看出,偏移量是一个带符号的 12 位二进制数,其取值范围为 $-2\,048 \sim 2\,047$ 。

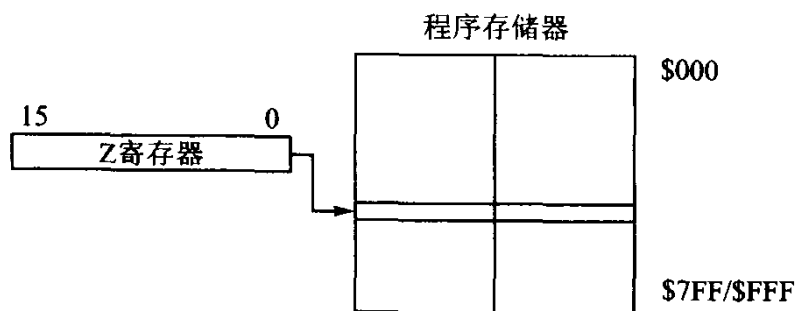


图 1.14 程序间接寻址

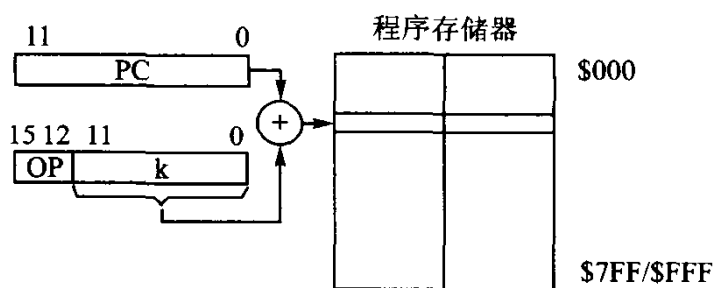


图 1.15 程序相对寻址

1.2.5 I/O 存储器

AT90S4414/8515 的所有 I/O 和涉外寄存器均被安放在了 I/O 空间。用输入(IN)和输出(OUT)指令访问这些 I/O 地址单元,就可在 I/O 空间和 32 个通用工作寄存器间进行数据传送。地址范围为 $\$00 \sim \$1F$ 的 I/O 寄存器均可按位寻址,即可用 SBI 指令把其中任何一位置 1,用 CBI 指令将之清 0,用 SBIS 和 SBIC 指令对这些寄存器的各位进行测试,看其为 1,抑或为 0。

表 1.2 中列出了 AT90S4414/8515 I/O 空间的所有寄存器。请注意,当使用特定的 I/O 指令 IN 和 OUT 访问这些寄存器时,必须用它们的 I/O 地址;而当把这些 I/O 寄存器作为 SRAM 单元访问时,则应该用它们的 SRAM 地址,即将 I/O 地址加 $\$20$ 。此表括号中的地址就是各寄存器的 SRAM 地址。

表 1.2 AT90S4414/8515 I/O 寄存器

地 址	符 号	名 称
$\$3F (\$5F)$	SREG	状态寄存器
$\$3E (\$5E)$	SPH	堆栈指针高字节