第二章 AVR 单片机系统结构

ATMEL 公司的 90 系列嵌入式单片微处理器是一种基于 AVR 增强性能、R ISC 结构的、低功耗、CMOS 技术、八位微控制器(Enhanced RISC Microcontrollers)，通常简称为 AVR 单片机。目前有 AT90S1200、AT90S2313、AT90S4414、AT90S8515、AT90S2323、AT90S4434、AT90S8535、ATmega83-163、ATmega603/103、ATmega161、ATtiny10/11/12/15/22/24/28 等多种型号。它们的基本结构都比较相近，这一章以 AT90S8515 单片机的内部结构为主来叙述 AVR 单片机系统结构。

2.1 AVR 单片机总体结构

90 系列单片机通过在单一时钟周期内执行功能强大的指令，每 MHz 可实现 1MIPS 的处理解力，这使得设计者可以优化功耗与速度之间的矛盾。

AVR 核为 32 个通用工作寄存器与丰富指令集的组合。32 个寄存器全部直接地与运算逻辑单元(ALU)连接，这使得可以通过在一个时钟周期内执行一条指令来访问到两个独立的寄存器。这种组合机构具备的代码效率比完成同样处理能力的常规 CISC 微控制器要快十倍。

90 系列单片机具有以下特征：1K～128K 字节可下载的 Flash 存储器，64～4K 字节 E2PROM，128～4K 字节 RAM，5～48 条通用的 I／O 线，32 个通用工作寄存器，带模拟比较器的定时器／计数器，可编程的异步 UART 串行口，内部及外部中断，带内部晶振的可编程看门狗定时器，一个为下载程序而设计的 SPI 串行口，10 位 A／D 转换器，以及 2 个可通过软件选择的省电模式。闲置模式停止 CPU 的工作，而 SRAM、定时器／计数器、片内振荡器(RTC)、SPI 口及中断系统继续工作。电源检测功能(BOD),掉电模式保留寄存器的内容，但冻结晶振，终止芯片的其它功能，直至下一次外部中断或硬件复位。

该器件的制造运用了 ATMEL 公司的高密度非易失存储器技术。芯片内可下载的 Flash 存储器可以通过 SPI 串行接口或通过通用的非易失存储器编程器对程序存储器进行系统内的重新编程。通过在单一芯片内将一个增强性能的 RISC 8 位 CPU 与可下载的 Flash 结合，使得 ATMEL 的 90 系列单片机成为一种适合于许多要求、具有高度灵活性和低成本的嵌入式控制应用的高效微控制器。

90 系列单片机 AVR 的全套编程和系统开发工具包括：C 编译器，BASCOM-AVR,宏汇编器，程序调试器／程序仿真器，系统在线仿真器和 SL-AVR 开发下载实验器。

图 2.1 为 AT90S8515 单片机方框图，说明了 90 系列单片机的内部结构。

一、引脚说明

AT90S4414/AT90S8515 引脚相同,仅 Flash、SRAM 和 E2PROM 相差一倍。AT90S4414/AT90S8515 引脚与 MCS-51 系列单片机 8X51/8X52 的引脚兼容,仅复位电平不同,AVR 低电平复位,MCS-51 高电平复位。这对用 AVR 单片机替代 MCS-51 单片机硬件电路带来方便。

(l) Vcc　Vcc 为供电引脚，连接到正电源。

(2) GND　GND 为接地引脚，连接到电源地。

(3) A 口(PA7～PA0) A 口为一个 8 位双向 I／O 口，每一引脚内部都有上拉电阻。A 输出缓冲器可以吸收 20mA 的电流，因而能直接驱动 LED 显示器。当 A 口被用于输入且内部上拉被触发时，如果外部被拉低，则会输出电流。当使用外部 SRAM 时，A 口作为复用的地址／数据和输入／输出口。

(4) B 口(PB7～PB0) B 口为一个 8 位双向 I／O 口，每一引脚内部都有上拉电阻。B 口的输出缓冲器可以吸收 20mA 的电流，当 B 口被用于输入且内部上拉被触发时，如果外部被拉低，则会输出电流。B 口也提供后面列出的 90 系列单片机许多特殊功能。

(5) C 口(PC7～PC0) C 口为一个 8 位双向 I／O 口，每一引脚内部都有上拉电阻。C 口的输出缓冲器可以吸收 20mA 的电流，当 C 口被用于输入且内部上拉被触发时，如果外部被拉低，则会输出电流。当使用外部 SRAM 时，C 口作为地址输出。

(6) D 口(PD7～PD0) D 口为带有内部拉高的 8 位双向 I／O 口。D 口的输出缓存器可以吸收 20mA 的电流。当 D 口被用于输入且内部上拉被触发时，如果外部被拉低，则会输出电流。D 口也提供后面列出的 90 系列单片机许多特殊功能。

(7) RESET　RESET 为复位输入。当晶振运行时，引脚上一个两周期的低电平可对器件进行复位。

(8) XTAL1　XTAL1 为晶振反相放大器的输入端和内部时钟操作电路的输入端。

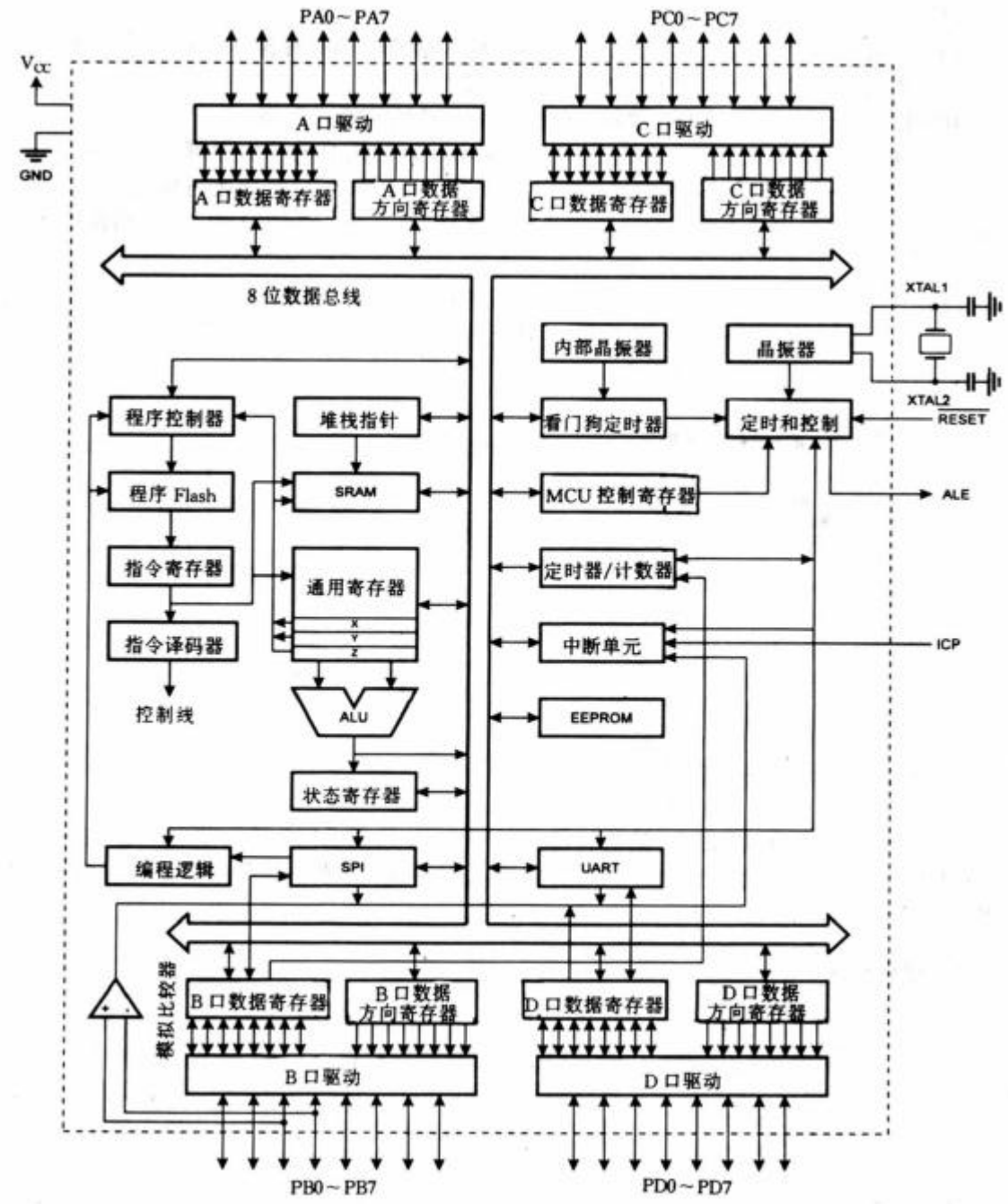


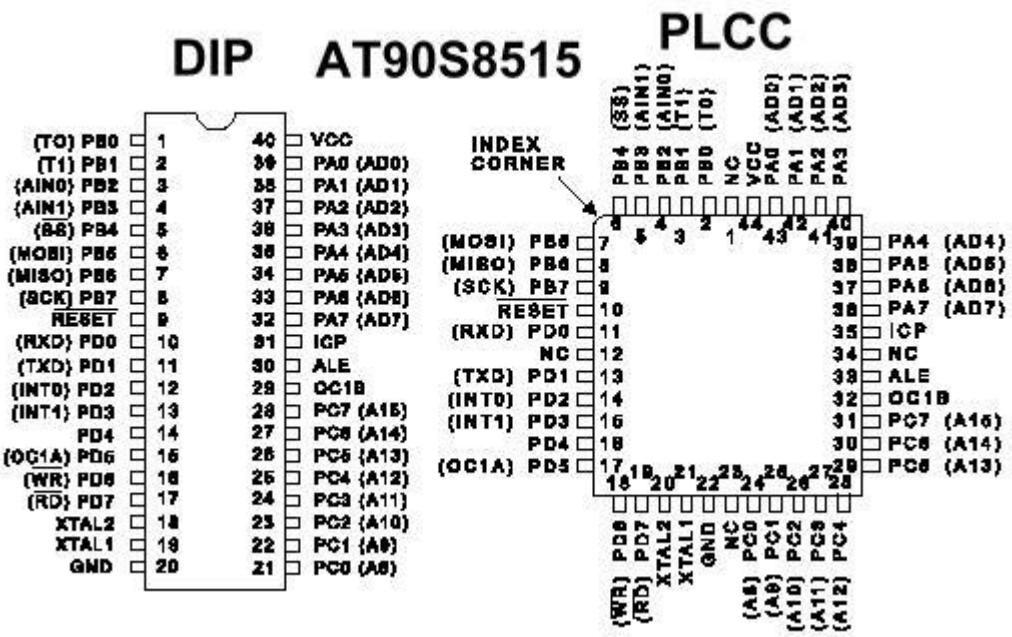
图 2.1　AT90S8515 单片机方框图

(9) XTAL2　XTAL2 以为晶振反相放大器的输出端。

(10) ICP　ICP 是定时器/计数器 l 的输入捕获功能的输入引脚。

(11) OC1B　OC1B 是定时器/计数器 1 的输出比较功能 B 的输出引脚。

(12) ALE　ALE 是使用外部存储器时的地址锁存触发端。ALE 选通门被用于在第一个访问周期中将低位地址锁存到地址锁存器中，而 PD0～PD7 在第二个访问周期中被用作数据。



二、晶振

XTAL1 和 XTAL2 单独地作为反相放大器的输入和输出，该放大器如图 2.2 所示，可被设置为片内的晶振。可使用石英晶振或陶瓷谐振器。为了由外部源驱动器件，当 XTAL1 被驱动时，XTAL2 不能连接，如图 2.3 所示。

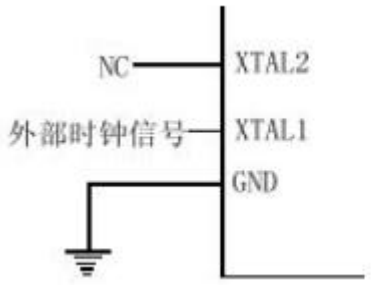


图 2.3　外部时钟驱动设置

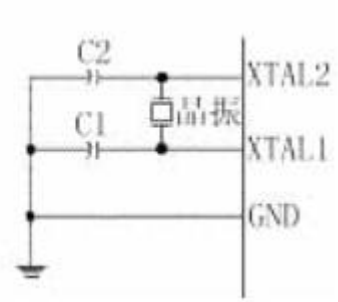


图 2.2　晶振连接

2.2 AVR 单片机中央处理器 CPU

90 系列单片机 AVR RISC 微控制器向上与 AVR 增强 RISC 结构兼容。为 90 系列单片机 MCU 而编写的程序与 AVR 8 位 MCU(AT90SXXXX)全部系列产品的源代码和运行的时钟周期相兼容。

2.2.1 结构概述

快速访问寄存器堆概念包括 32 个带有单一时钟周期访问时间的 8 位通用工作寄存器。这意味着在一个单一时钟周期内，执行一个 ALU 操作(运算逻辑单元)。从寄存器堆中输出两个操作数，且操作数被执行，运行结果被存储回寄存器堆——这些操作在一个时钟周期内完成。

32 个寄存器中的 6 个寄存器作为 3 个 16 位间接地址寄存器指针，被用于数据空间寻址，从而可以进行高效的地址计算。3 个寄存器中的一个还被用作地址指示器，完成常量表的查询功能。这些新加的功能寄存器为 16 位 X—寄存器，16 位 Y—寄存器，16 位 Z—寄存器。

ALU 支持寄存器之间的运算和逻辑功能，以及常数和寄存器之间的运算与逻辑功能。ALU 也执行单一的寄存器操作，图 2.4 所示为 AT90S8515 单片机 AVR 的结构。

除了寄存器操作功能之外，常规存储器寻址模式还可用在寄存器堆上。寄存器堆被分配在最低的 32 个数据空间地址($00～$1F)，当触发时，即使它们为一般的存储地址，也能访问到。

I／O 存储器空间包含 64 个作为 CPU 外围功能的地址,为控制寄存器、定时器／计数器、A／D 转换器及其它 I／O 功能。I／O 存储器可被直接访问，或作为寄存器堆之后的数据空间，地址为 $20～$5F。

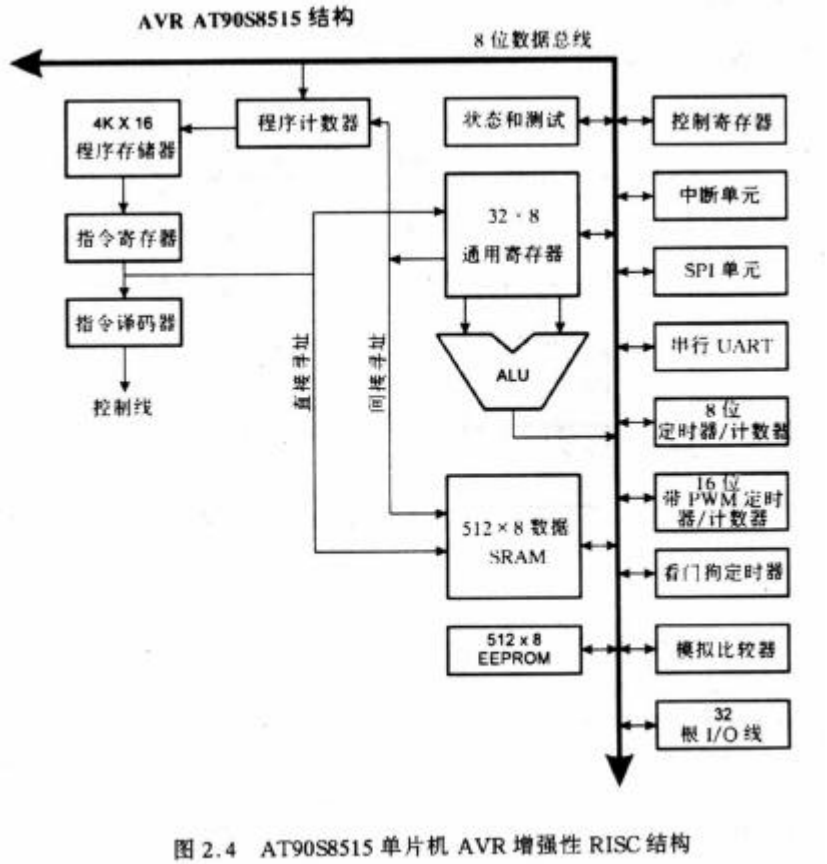
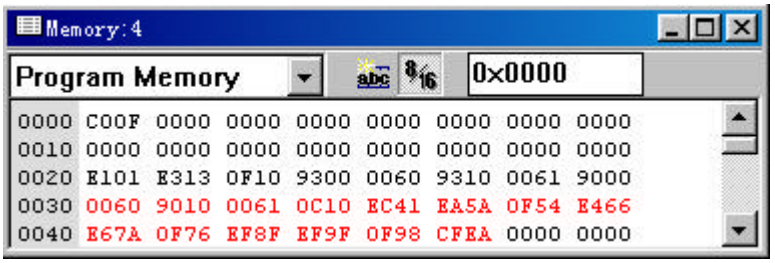


图 2.4　AT90S8515 单片机 AVR 增强性 RISC 结构

AVR 运用 Harvard 结构概念，即对程序存储和数据带有不同的存储器和总线。通过单一级的流水线可对程序存储器进行访问。当执行某一指令时，下一指令被预先从程序存储器中取回。这

使得指令可以在每一个时钟周期内被执行。芯片内的程序存储器为系统内可下载的 Flash 存储器。

通过相关的转移和调用指令，全部的 4K 地址空间可以被直接访问到。所有 AVR 指令均有一个单一的 16 位字格式，这意味着每个程序存储器地址包含了一个单一的 16 位或 32 位指令。



在中断和子程序调用过程中，返回地址程序计数器(PC)被存储于堆栈之中。该堆栈被高效率地放置在通用 SRAM 中，作为结果，堆栈的大小只被 SRAM 的大小及对 SRAM 的使用而限制。所有的用户程序必须在复位状态初始化 SP(在子程序和中断程序被执行之前,因为复位后,SP 为 0000H)。16 位的堆栈指针 SP 可在 I/O 空间之中被进行读/写访问。

128B～4K 字节的数据 SRAM 可以通过 AVR 结构中的不同种寻址模式很容易地访问。

AVR 结构中的存储器空间均为线性和有规律的存储器映射。图 2.5 为存储器映射图。

一个灵活的中断模块，在 I/O 空间中有它自己的控制寄存器，并且在状态寄存器中带有一个附加的全局中断触发位。所有不同的中断，在程序存储器开始位置的中断向量(对应固定中断入口地址及中断申请名称)表中，带有一个独立分开的中断向量。不同的中断优先级与中断向量位置相一致。中断地址向量的位置越低，优先级越高(中断向量控制方式同 MCS-51 单片机)。

2.2.2 通用寄存器堆

图 2.6 为 AVR 单片机 AT90S8515 CPU 中 32 个通用寄存器的结构图。

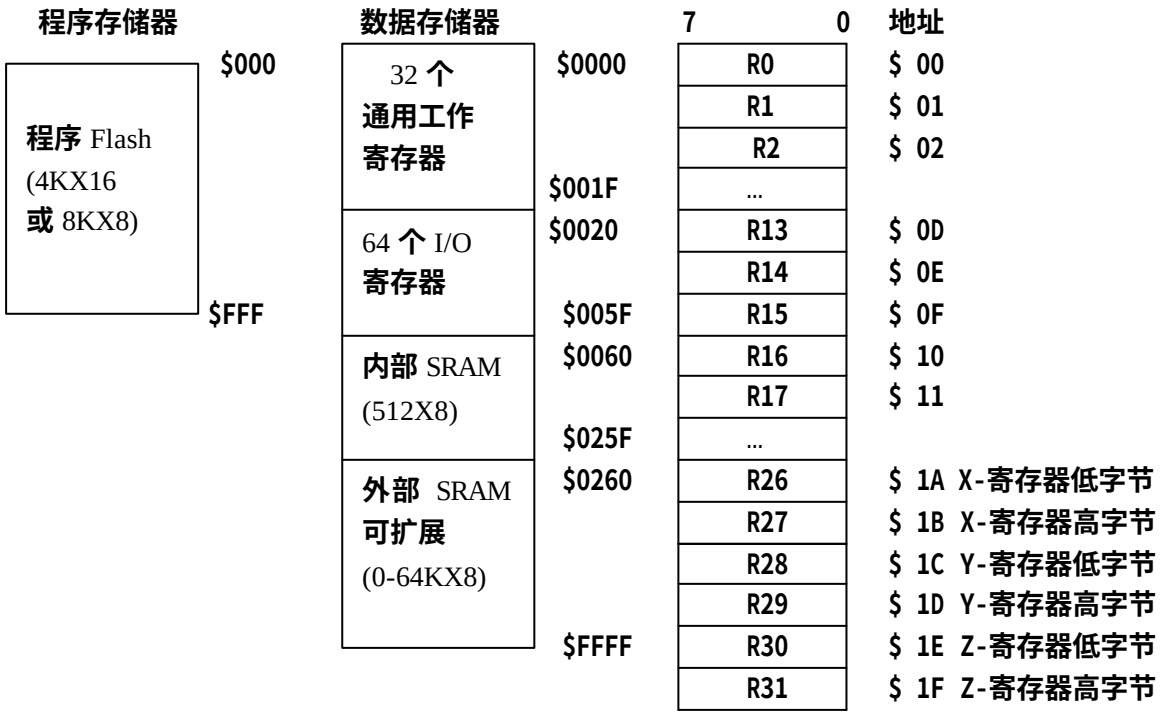


图 2.5 存储器映射图 (AT90S8515)　　图 2.6 AVR 单片机 CPU 通用寄存器

指令集中所有的寄存器操作指令均带有方向，并能在单一周期中访问所有的寄存器。唯一的

例外是，存在于常量和寄存器之间的 5 个常量运算和逻辑指令 SBCI(带 C 减立即数)、SUBI(减立即数)、CPI(与立即数比较)、ANDI(与立即数)、ORI(或立即数)以及直接装入立即数的 LDI 指令。这些指令适用于寄存器堆（ R16～ R31）之中后半部分的寄存器。通用的 SBC(带进位减)、SUB(减法)、CP(比较)、AND(与)、OR(或)指令，以及两个寄存器之间的操作指令或单一寄存器操作指令，在整个寄存器堆中（ R0 ～ R31）都是适用的。

如图 2.6 所示，每个寄存器被分配给一个数据存储器地址，将其直接映射到用户数据空间的前 32 个地址。虽然寄存器堆并未像 SRAM 定位那样提供物理地址，但寄存器 X、 Y、Z 可被设置用来对寄存器堆中的寄存器做索引，这种存储器的组成结构在访问寄存器时提供了极大的灵活性。

2.2.3 X、Y、Z 寄存器

为了实现通用目的，寄存器 R26 ～ R31 具有一些新加的功能。这些寄存器作为对数据空间间接寻址的地址指针。这三个间接寄存器 X、Y、Z 由图 2.7 定义。在不同的寻址模式下，这些地址寄存器的功能为固定位移、自动增量和减量（请参考不同的指令）。

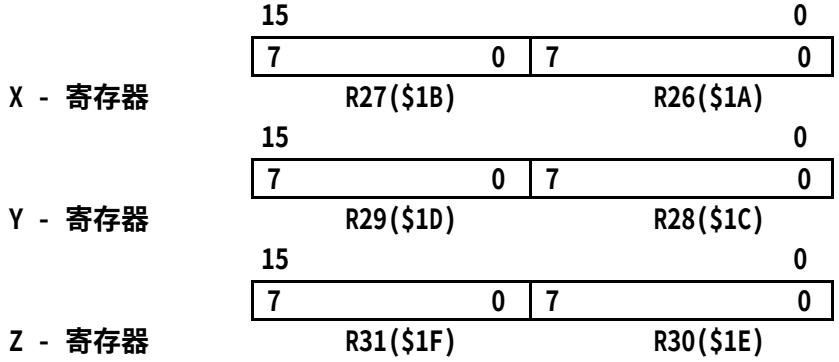


图 2.7　X、Y、Z 寄存器

2.2.4　ALU 运算逻辑单元

高性能的 AVR 运算逻辑单元的操作与所有的 32 个通用寄存器保持直接的连接。在单一时钟周期内，ALU 是在寄存器堆中的寄存器之间执行操作。ALU 操作被分为 3 个主要的目的：算法、逻辑和位功能。AVR 产品家族中 ATmega161/L 微控制器的 ALU 运算部件中,有硬件乘法器。

2.3 AVR 单片机存储器组织

2.3.1　可下载的 Flash 程序存储器

90 系列单片机包括 1K～ 128K 字节的片内可下载 Flash 存储器。由于所有指令为 16 位字或 32 位双字，用于存储程序的 Flash 的结构为（512～64K 字） X 16。 Flash 存储器的使用寿命最少为 1000 次写 / 擦循环。AT90S8515 单片机程序计数器宽为 12 位，以此来对 4K 个程序存储器地址寻址。

常量表必须被设定在 0～ 4K 的地址之间（请参考 LPM——装载程序存储器指令说明）。

2.3.2　内部和外部的 SRAM 数据存储器

图 2.8 说明 AT90S8515 单片机的数据存储器组成。低端 608 个数据存储器地址编址为寄存器堆、I / O 存储器和内部数据 SRAM。前 96 个地址编址为寄存器堆 + I / O 存储器，接下来的 512 个地址是内部数据 SRAM。可选的外部数据 SRAM 可以放置在同一个 SRAM 空间中，该 SRAM 将占据内部 SRAM 之后的地址直到 64K 减 1，这由 SRAM 的大小来定。

当访问超出内部数据 SRAM 地址的数据存储器空间时就访问外部数据 SRAM，使用与访问内部数据 SRAM 同样的指令。当访问内部的 SRAM 时，读写控制信号（/RD 和/WR）在整个访问周期中是非触发的，外部的 SRAM 的操作通过设置 MCUCR 寄存器的 SRE 位来触发，详见后面 MCU 控制寄存器的说明。

访问外部 SRAM 存储器比内部 SRAM 多用一个时钟周期。使用命令 LD、ST、LDS、STS、PUSH、POP 能访问外部 SRAM 存储器。如果堆找被放置在外部 SRAM 中，则中断程序调用和返回指令将多用 2 个时钟周期。当使用带有等待状态的外部 SRAM 时，外部的 SRAM 将额外花费 4 个时钟周期。

图 2.8　SRAM 组织

对数据存储器的 5 个不同寻址模式为：直接、带位移的间接、间接、带预减量的间接和带后增量的间接寻址。在寄存器堆中，寄存器 R26～R31 具有间接寻址指针寄存器的特性。间接寻址到达数据地址空间的尽头。带位移的间接寻址模式的特性为，它可到达由寄存器 Y 和 Z 给出的基本地址的 63 个地址位。当使用自动预减量和后增量的间接寻址模式时，地址寄存器 X、Y 和 Z 被使用，或被增大、减小。

32 个通用工作寄存器、64 个 I/O 寄存器，以及 AT90S8515 单片机中的 512 字节数据 SRAM 可通过所有这些地址模式被直接访问到。

2.3.3　E2PROM 数据存储器

90 系列单片机包括 64～4K 字节的 E2PROM 存储器。它被组织为一个分开的数据空间，这个数据空间用单字节可被读写。E2PROM 的使用寿命至少为 100000 次写/擦循环。在 E2PROM 和 CPU 之间的访问详见后面对 E2PROM 的地址寄存器、数据寄存器和控制寄存器的说明。

2.3.4　存储器访问和指令执行时序

本节说明了 90 系列单片机指令执行和内部存储器访问的时序。

AVR CPU 由系统时钟Φ驱动，直接由芯片的外部时钟晶振触发，没有使用内部时钟分频。

图 2. 9 所示为 Harvard 结构和快速访问寄存器堆概念触发的并行指令存取和指令执行

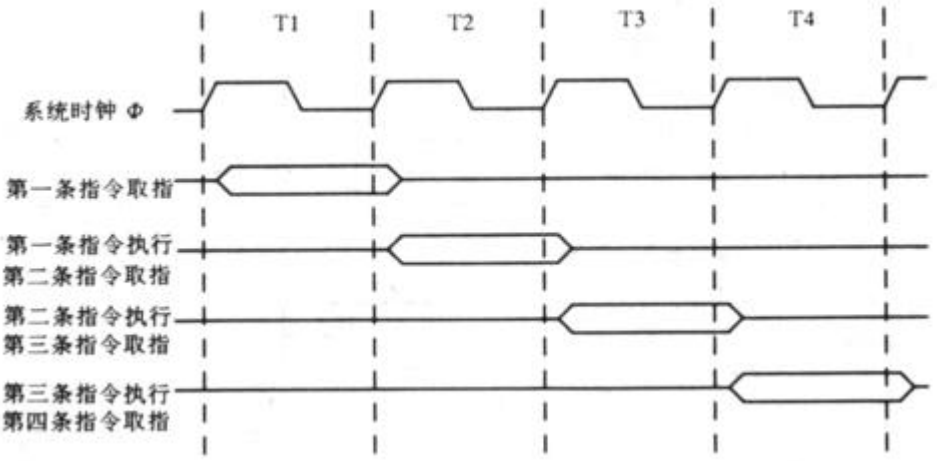


图 2.9 并行指令存取和指令执行。

时序。这种基本的流水线概念目的是为了获得高达每 1 MIPS/MHz 的效率。

图 2.10 所示为寄存器堆的内部定时概念。在单一时钟周期内，使用 2 个寄存器操作数的一个 ALU 操作被执行，而其结果被存储回目的寄存器。

图 2.11 所示为在 2 个系统时钟周期内，完成内部数据 SRAM 的访问。

图 2.12 所示为在 2 个系统时钟周期内，完成外部数据 SRAM 的访问。

图 2.13 所示为含有等待状态的（等待状态触发）外部数据 SRAM 的访问时序。

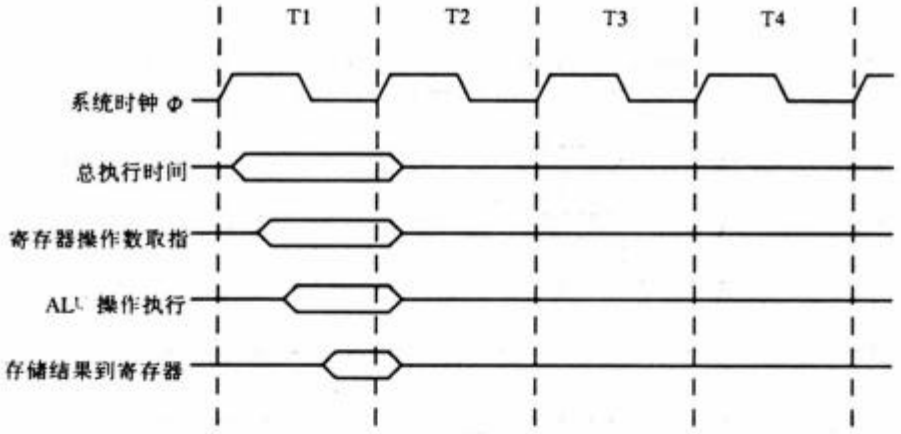


图 2.10　单周期 ALU 操作

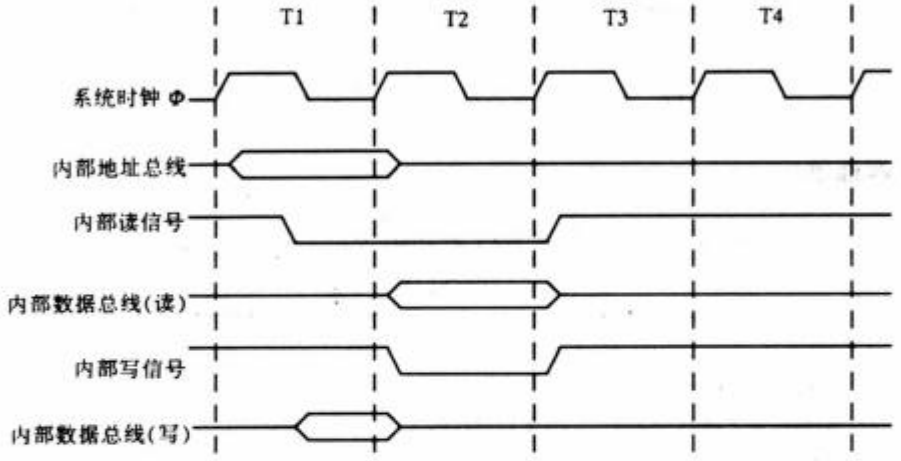


图 2.11　片内数据 SRAM 访问周期

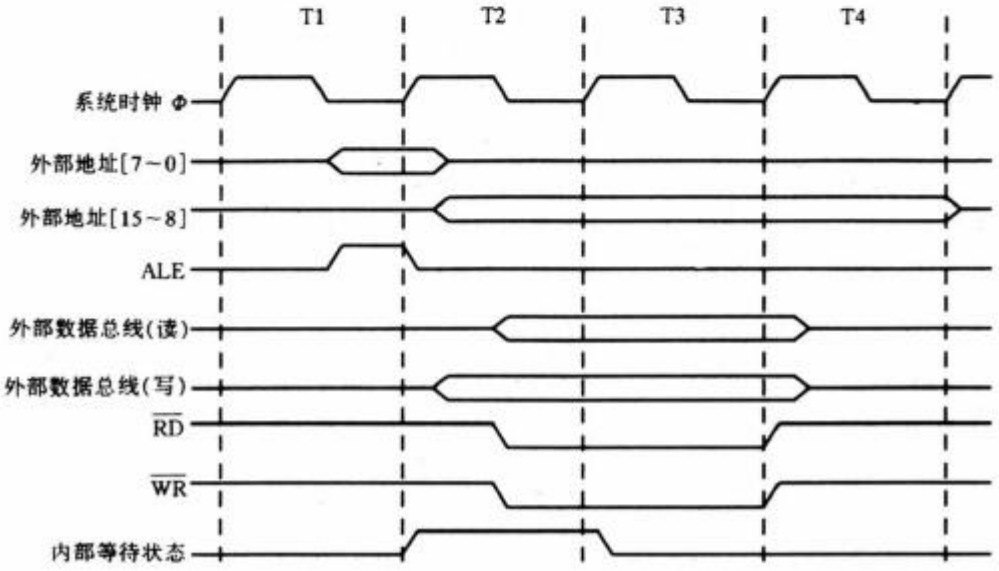


图 2.12　无等待状态的外部数据 SRAM 访问周期

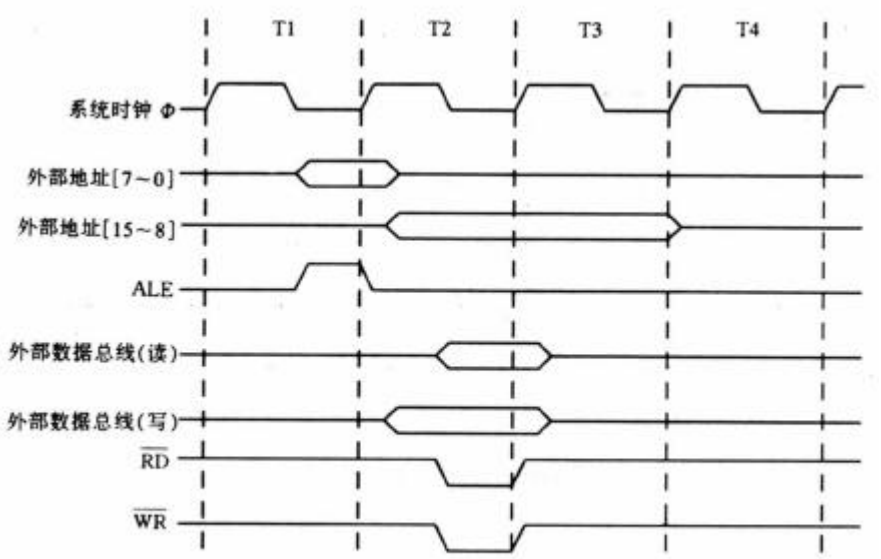


图 2.13　含有等待状态的外部数据 SRAM 访问周期

2.3.5　I/O 存储器

在编写源文件时一定要写该器件的配置文件,作为源文件的文件头,如你选用 AT90S8515 **单片机,源文件的文件头为:**

.include"8515def.inc"　;文件头就是该器件的 I/O **寄存器及位地址的定义文件,汇编时用到它。可查看光盘中** “\AVR\asmpack\appnotes*.inc”**文件**

表 2.1 所示为 AT90S8515 单片机的 I/O 空间定义。

表 2.1　AT90S8515 I/O 空间

十六进制地址	名　称	功　能
$3F($5F)	SREG	状态寄存器
$3E($5E)	SPH	堆栈指针高位
$3D($5D)	SPL	堆栈指针低位
$3B($5B)	GIMSK	通用中断屏蔽寄存器
$3A($5A)	GIFR	通用中断标志寄存器
$39($59)	TIMSK	定时器/计数器中断屏蔽寄存器
$38($58)	TIFR	定时器/计数器中断标志寄存器
$35($55)	MCUCR	MCU 通用控制寄存器
$33($53)	TCCR0	定时器/计数器'0 控制寄存器
$32($52)	TCNT0	定时器/计数器 0(8 位)
$2F($4F)	TCCR1A	定时器/计数器 1 控制寄存器 A
$2E($4E)	TCNT1B	定时器/计数器 1 控制寄存器 B
$2D($4D)	TCNT1H	定时器/计数器 1 高字节
$2C($4C)	TCNT1L	定时器/计数器 1 低字节
$2B($4B)	OCR1AH	定时器/计数器 1 输出比较寄存器 A 高字节
$2A($4A)	OCR1AL	定时器/计数器 1 输出比较寄存器 A 低字节
$29($49)	OCR1BH	定时器/计数器 1 输出比较寄存器 B 高字节
$28($48)	OCR1BL	定时器/计数器 1 输出比较寄存器 B 低字节
$25($45)	ICR1H	T/C1 输入捕获寄存器高字节

表 2.1 续

十六进制地址	名　称	功　能
$24($44)	ICR1L	T/C1 输入捕获寄存器低字节
$21($41)	WDTCR	看门狗定时控制寄存器
$1F($3F)	EEARH	EEPROM 地址寄存器高字节
$1E($3E)	EEARL	EEPROM 地址寄存器低字节
$1D($3D)	EEDR	EEPROM 数据寄存器
$1C($3C)	EECR	EEPROM 控制寄存器
$1B($3B)	PORTA	A 口数据寄存器
$1A($3A)	DDRA	A 口数据方向寄存器
$19($39)	PINA	A 口输入脚
$18($38)	PORTB	B 口数据寄存器
$17($37)	DDRB	B 口数据方向寄存器
$16($36)	PINB	B 口输入脚
$15($35)	PORTC	C 口数据寄存器
$14($34)	DDRC	C 口数据方向寄存器
$13 ($33)	PINC	C 口输入脚
$12 ($32)	PORTD	D 口数据寄存器
$11($31)	DDRD	D 口数据方向寄存器
$10($30)	PIND	D 口输入脚
$0F($2F)	SPDR	SPI I/O 数据寄存器
$0E($2E)	SPSR	SPI 状态寄存器
$0D($2D)	SPCR	SPI 控制寄存器
$0C($2C)	UDR	UART I/O 数据寄存器
$0B($2B)	USR	UART 状态寄存器
$0A($2A)	UCR	UART 控制寄存器
$09($29)	UBRR	UART 波特率寄存器
$08($28)	ACSR	模拟比较控制和状态寄存器

注意：保留的和未用到的地址未列。

你打开器件配置文件(*.inc)查看一下,防止没有器件配置文件头汇编时出错,有了器件配置文件头,在编写源程序时就不必重复定义 I/O 口及位地址等!

90 系列单片机所有不同的 I/O 口和外围设备均在 I/O 空间中已设置好。不同的 I/O 地址可以通过 IN 和 OUT 指令访问，这些指令在 32 个通用寄存器与 I/O 空间之间传输数据。地址范围$00-$1F 之间的寄存器可通过 SBI(I/O 寄存器置$FF)和 CBI(I/O 寄存器清零)指令直接一位一位的访问。使用 SBIS(寄存器位置 1)和 SBIC(寄存器位清零)指令对这些寄存器中的每一位值进行检验。请参考指令系统一章。

当使用 I/O 特定的命令时，必须使用 IN(I/O 口输入)、OUT(输出到 I/O 口)、SBIS(I/O 位置位跳行)和 I/O 地址的$00~$3F。当寻址的 I/O 寄存器为 SRAM 时，必须向该地址加入$20。本资料中的全部 I/O 寄存器地址均带有列在圆括号中的 SRAM 地址。以下部分说明了不同 I/O 和

外围设备的控制寄存器。

一、状态寄存器——SREG

AVR 的状态寄存器 SREG 在 I／O 空间的地址为$3F（$5F），定义如下：

位	7	6	5	4	3	2	1	0	
$3F($5F)	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	
初始化值	$00								

状态寄存器 SREG 的作用很大，不用位代表不同的含义,对某位的操作,置 1 或清零,反映运算、操作结果状态,有置 1,清零,为 1 转移,为 0 转移等,状态寄存器 SREG 有 36 条指令供使用。

位 7--I：全局中断触发

全局中断触发位必须被设置（1），以便允许中断，在这之后，单独的中断触发控制在中断屏蔽寄存器 GIMSK／TIMSK 中完成。如果全局中断触发寄存器被清空（0），所有中断被禁止，GIMSK／TIMSK 值独立存在。在中断发生后，I 位由硬件清除，并由 RETI(中断返回)指令设置，从而允许子序列的中断。

位 6—T：位复制存储

位复制指令 BLD(SREG 中 T 标志到寄存器某位)和 BST(寄存器某位到 SREG 中 T 标志)使用 T 位作为源操作位和目标操作位。寄存器堆中某一寄存器的某一位可以通过 BST 指令被复制到 T，用 BLD 指令则可将 T 中的位值复制到寄存器堆中的某一寄存器的某一位。

位 5—H：半进位标志位

半进位标志位 H 指示了在一些运算操作过程中的半进位(低四位向高四位进位)，请参考指令集说明。

位 4—S：标志位,S=N+V

S 位是负数标志位 N 和 2 的补码溢出标志位 V 两者异或值，请参考指令集说明。

位 3—V：2 补码溢出标志位

2 的补码溢出标志位 V 支持 2 的补码运算，请参考指令集说明。

位 2—N：负数标志位

负数标志位 N 指示在不同的运算和逻辑操作之后的负数结果，请参考指令集说明。

位 1—Z：零值标志位

零值标志位 Z 指示在不同的运算和逻辑操作之后的零值结果，请参考指令集说明。

位 0—C：进位标志位

进位标志位 C 指示在某一运算和逻辑操作中的某一进位，请参考指令集说明。

二、堆栈指针——SP

在 I／O 地址$3E（$5E）和$3D（$5D）的两个 8 位寄存器构成了 90 系列单片机的 16 位堆栈指针。由于 AT90S8515 单片机支持 64K 字节的 SRAM,所以所有 16 位都被使用。

位	15	14	13	12	11	10	9	8	
$3E($5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
$3D($5D)	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	
读／写	R／W	R／W	R／W	R／W	R／W	R／W	R/W	R/W	
初始化值	$00	00							

因为复位后堆栈为 SPH=$00,SPL=$00,AVR 单片机堆栈是-1 或-2 进栈,所以主程序一开始堆栈指针必须设在 SRAM 最高处,如 AT90S8515 的堆栈设在 SRAM 的$025F 处。堆栈有自动进栈(调用指

令、中断指令)和人工进栈指令有压栈(PUSH)、出栈(POP)指令。

堆栈指针指示了数据 SRAM 堆栈区域，子程序和中断堆栈被放置在该区域中。在数据 SRAM 中的该堆栈空间必须在执行任何子程序调用或中断触发之前被程序定义。当执行 PUSH 指令，数据被压入堆栈时，堆栈指针减少 1。当执行子程序 CALL 和中断而将数据压入堆栈时，堆栈指针减少 2。当执行 POP 指令而数据从堆栈弹出时，堆栈指针增大 1。当从子程序 RET 返回或从中断 RETI 返回数据被从堆栈弹出时，堆钱指针增大 2。

2.4　AVR 单片机系统复位

90 系列单片机提供多种不同的中断源。这些中断和与之分开的复位向量均在程序存储器空间中各自带有分开的程序向量地址。所有中断均分配给单独的触发位，这些触发位须与状态寄存器中的 I 位一起被设为 1，以便能执行中断。

程序存储器空间中最低的地址被自动地定义为复位和中断向量。如表 2. 2 所列，表中还列出了不同中断的优先级。地址越低，优先级越高。/RESET 有最高的优先级,以下依次为 INT0…，即外部中断请求 0 等。

表 2.2　复位和中断向量

向量号	程序地址	源	中　断　定　义
1	$ 000	$\overline{\text{RESET}}$	硬件脚和看门狗复位
2	$ 001	INT0	外部中断请求 0
3	$ 002	INT1	外部中断请求 1
4	$ 003	TIMER1 CAPT	定时器/计数器 1 捕获事件
5	$ 004	TIMER1 COMPA	定时器/计数器 1 比较匹配 A
6	$ 005	TIMER1 COMPB	定时器/计数器 1 比较匹配 B
7	$ 006	TIMER1 OVF	定时器/计数器 1 溢出
8	$ 007	TIMER0 OVF	定时器/计数器 0 溢出
9	$ 008	SPI, STC	串行传送完成
10	$ 009	UART, RX	UART, RX 完成
11	$ 00A	UART, UDRE	UART 数据寄存器空
12	$ 00B	UART, TX	UART, TX 完成
13	$ 00C	ANA _ COMP	模拟比较器

以下为复位和中断向量地址的典型和通用的程序设置:

```
地 址        标 号    代  码                    注释
$ 000                 RJMP    RESET          ; 复位处理
$ 001                 RJMP    EXT__INT0      ; 转 INT0 处理
$ 002                 RJMP    EXT__INT1      ; 转 INT1 处理
$ 003                 RJMP    TIMI__CAPT     ; 转定时器 1 捕获处理
$ 004                 RJMP    TIMI__COMPA    ; 转定时器 1 比较 A 处理
$ 005                 RJMP    TIMI__COMPB    ; 转定时器 1 比较 B 处理
$ 006                 RJMP    TIMI__OVF      ; 转定时器 1 溢出处理
$ 007                 RJMP    TIMO__OVF      ; 转定时器 0 溢出处理
$ 008                 RJMP    SPI__HANDLE    ; 转 SPI TX 处理
```

```
$ 009          RJMP    UART__RXC   ; 转 UART,RX 完成处理
$ 00A          RJMP    UART__DRE   ; 转 UART,UDR 空处理
$ 00B          RJMP    UART__TXC   ; 转 UART,TX 完成处理
$ 00C          RJMP    ANA__COMP   ; 转模拟比较器处理
;不用的中断入口地址写上 RETI,有抗干扰作用
$ 010     MAIN: <instr>  X  X  X    ; 主程序开始，主程序必须跳过中断区，
...              ...      ...
```

2.4.1　复位源

90 系列单片机有 3 个复位源：

- **加电复位**。当供电电平加至 VCC 和 GND 引脚时，MCU 进行复位。
- **外部复位**。当一个低电平加到/RESET 引脚多于 2 个 XTAL 周期时，MCU 进行复位。
- **看门狗复位**。当看门狗定时器超时，且看门狗为触发时，MCU 进行复位。

在复位过程中，所有的 I／O 寄存器被设为初始值，程序从地址$0 000 开始执行。$0000 地址中放置的指令须为某一 RJMP——相对转移，即到达复位处理路径的指令。若程序从没有对中断源触发，则中断向量无法使用，正常的程序代码可以放置在这些地址中。图 2. 14 的电路图说明了复位逻辑。表 2.3 定义了复位电路的时序和电参数。

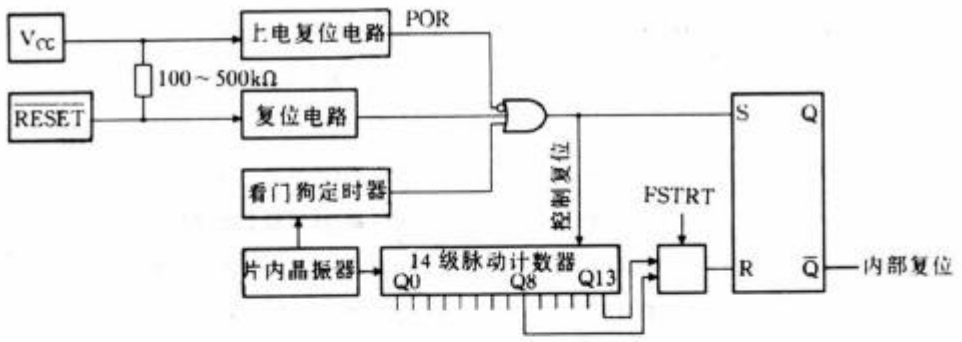


图 2.14　复位逻辑

表 2.3　复位特性(V_{CC} = 5.0 V)

符　号	参　　数	最小值	典型值	最大值	单　位
V_{POT}	上电复位门限电压	1.8	2	2.2	V
V_{RST}	复位脚门限电压		V_{CC}/2		V
t_{POR}	上电复位周期	2	3	4	ms
t_{TOUT}	复位延时定时输出周期 FSTRT 不编程	11	16	21	ms
t_{TOUT}	复位延时定时输出周期 FSTRT 编程	1.0	1.1	1.2	ms

2.4.2　加电复位

加电复位（POR）电路确保了只有当 Vcc 达到一个安全电平时，器件才开始工作。如图 2. 15 所示，当看门狗定时器晶振对内部定时器定时时，不启动 MCU，直到 Vcc 到达 Power_on 门槛电压—V_{POT} 一定时间之后才启动 MCU（如图 2.16 和 2.17 所示）。全部的复位时间为 Power_on 复位时间—t_{POR} 加上延时时间 t_{TOUT}。

如果使用了陶瓷谐振器或其它快速启动的晶振来为 CPU 定时，Flash 中的 FSTRT 熔丝位可被编程来给出一个更短的启动时间。由于片内电阻将/RESET 拉高，在无需外部复位时，引脚不连接。将/RESET 与 Vcc 连接，则产生同样效果。在对 Vcc 加电一段时间后，通过将/RESET 引脚拉低，加电复位时间可以被延长。请参考图 2. 17 的时序例子。

2. 4. 3 外部复位

/RESET 复位引脚上的低电压引发外部复位。该引脚必须被拉低至少 2 个晶振时钟周期，

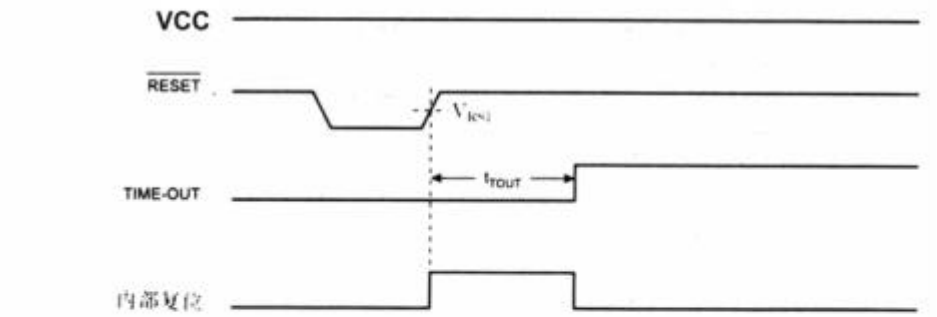


图 2.18　在操作期间由外部复位

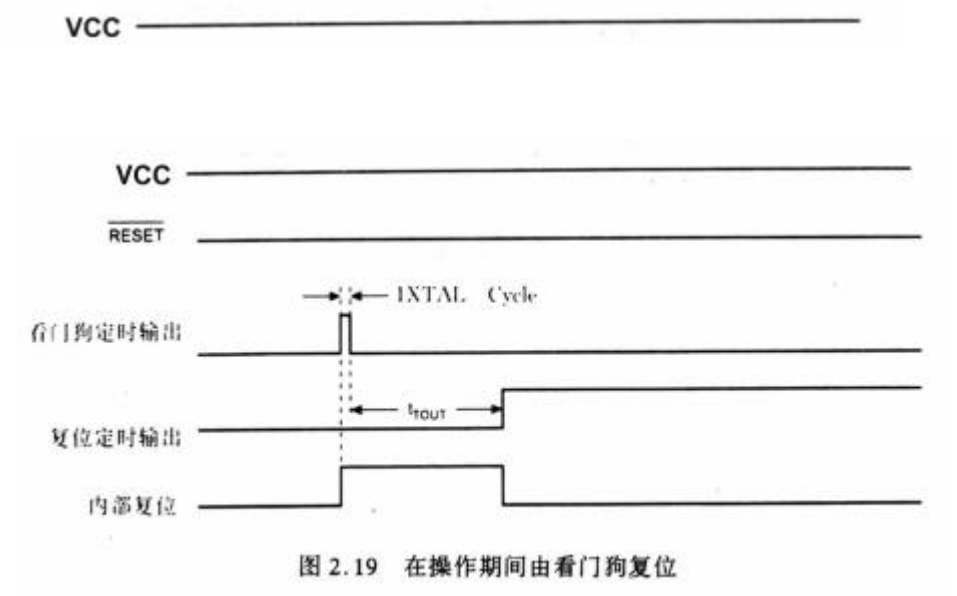


图 2.19　在操作期间由看门狗复位

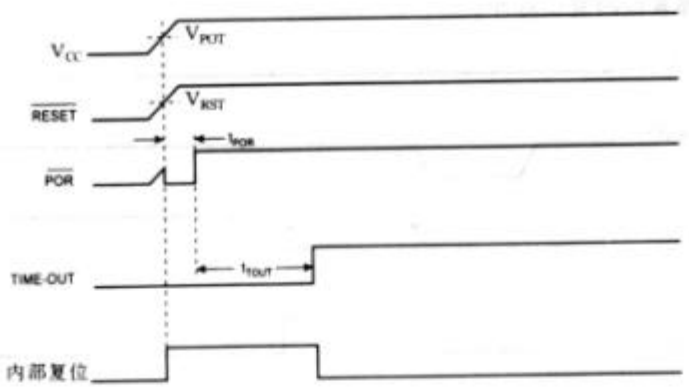


图 2.15　MCU 启动，$\overline{\text{RESET}}$连或不连到 V_{CC}，V_{CC}快速上升

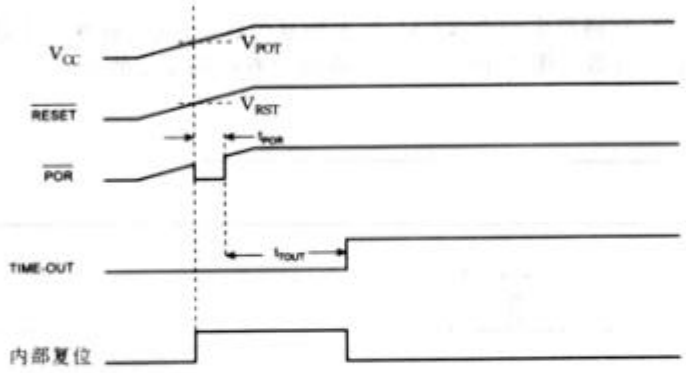


图 2.16　MCU 启动，$\overline{\text{RESET}}$连或不连到 V_{CC}，V_{CC}慢速上升

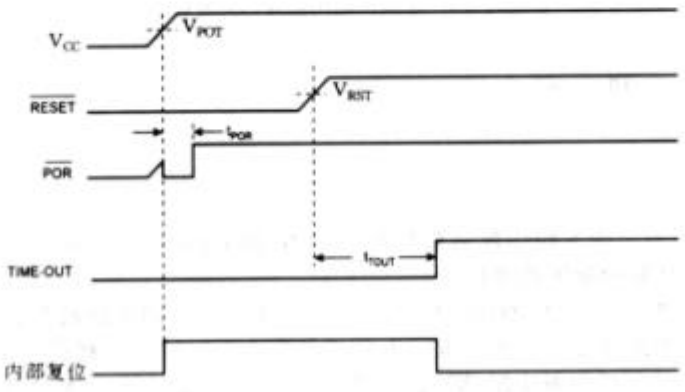


图 2.17　MCU 启动，$\overline{\text{RESET}}$由外部控制

在 t_{TOUT}周期结束后，当在其正边缘达到复位门槛电压时，延迟定时器启动 MCU。图 2.18 为

在操作期间由外部复位的复位脉冲。

2.4.4 看门狗复位

看门狗定时输出时，它将产生一个 XTAL 的短暂复位脉冲，在此脉冲的下降沿，延迟定时器开始对超时时间 t_{TOUT} 计数。图 2.19 为在操作期间由看门狗复位的复位脉冲。

2.5　AVR 单片机中断系统

2.5.1　中断处理

90 系列单片机有 2 个 8 位中断屏蔽控制寄存器，即 GIMSK——通用中断屏蔽寄存器和 TIMSK——定时／计数中断屏蔽寄存器。

当某一中断发生时，全局中断触发位 I 位被清空（0），则全部中断被禁止。用户软件可以将 I 位置为 1 来对中断触发。当执行 RETI，从中断指令返回时，I 位被设为 1。

由于维持静态的事件引发中断（即输出比较寄存器 1 与定时器／计数器 1 的值相匹配），当事件发生时,中断标志被设置为 1。若中断位被清空,且中断条件维持原状,标志直到下次事件发生时才被设置为 1。

当程序计数器指向现行中断时，执行中断处理程序，硬件将相应的产生中断的标志位清空。

一、通用中断屏蔽寄存器——GIMSK

位	7	6	5	4	3	2	1	0	
$3B($5B)	INT1	INT0	-	-	-	-	-	-	GIMSK
读／写	R／W	R／W	R	R	R	R	R	R	
初始化值	$00								

位 7—INT1：　外部中断请求 1 触发

当 INT1 被置为 1，且状态寄存器（SREG）中 I 位被置为 1 时，外部引脚中断被触发。MCU 通用控制寄存器（MCUCR）中的中断检测控制 1 位 I／0（ISCll 和 ISC10）定义的外部中断，在 INT1 引脚上是在电平还是上升或下降边沿被触发。即使 INT1 被定义为输出，引脚上的事件也将引发一个中断请求。外部中断请求 1 的中断从程序存储器地址$002 中被执行。

位 6—INT0:外部中断请求 0 触发

当 INT0 被设为 1，且状态寄存器（SREG）中 I 位被设为 1 时，外部引脚中断被触发。MCU 通用控制寄存器（MCUCR）中的中断检测控制 0 位 I／0（ISC01 和 ISC00）定义的外部中断，在 INT0 引脚上是在电平还是上升或下降边沿被触发。即使 INT0 被定义为输出，引脚上的事件也将引发一个中断请求。外部中断请求 0 的中断从程序存储器地址$001 中被执行。

位 5～0—Res：保留位

90 系列单片机的该位为保留位，总读 0。

二、通用中断标志寄存器——GIFR

位	7	6	5	4	3	2	1	0	
$3A($5A)	INTF1	INTF0	-	-	-	-	-	-	GIFR
读／写	R／W	R／W	R	R	R	R	R	R	
初始化值	$00								

位 7—INTF1：外部中断标志 1

当 INT1 引脚上的某一事件触发了某一中断请求，INTF1 置为设 1。若 SREG 中的 I 位以及 GIMSK 中的 INT1 位被置 1，MCU 将在地址$002 向中断向量转移。当中断程序执行时，标志被清空。标志位还可以通过向其写入逻辑 0 来清空。

位 6—INTF0：外部中断标志 0

当 INT0 引脚上的某一事件触发了某一中断请求，INTF0 变为置 1。若 SREG 中的 I 以及 GIMSK 中的 INT0 位被设为了 1，MCU 将在地址$002 向中断向量转移。当中断程序执行时，标志被清空。标志位还可以通过向其写入逻辑 0 来清空。

位 5～0—Res：保留位

90 系列单片机的该位为保留位，总读 0。

三、定时器／计数器中断屏蔽寄存器——TIMSK

位	7	6	5	4	3	2	1	0	
$39($59)	TOIE1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE0	-	TIMSK
读／写	R／W	R／W	R/W	R	R/W	R	R/W	R	
初始化值	$00								

位 7–TOIE1:定时器/计数器 1 溢出中断触发

当 TOIE1 被设为 1，且状态寄存器中 I 位被设为 1 时，定时器／计数器 1 溢出中断触发。如定时器／计数器 1 产生溢出，相应的中断（在向量$006）被执行。在定时器／计数器中断标志寄存器——TIFR 中的溢出标志（定时器 1）被设置为 1。当定时器／计数器 1 处于 PWM 模式下，计数器在$0000 变化计数方向时，定时器溢出标志被设置。

位 6–OCIE1A：定时器/计数器 1 输出比较匹配 A 中断触发

当 OCIE1A 被设为 1，且状态寄存器中的 I 位被设为 1 时，定时器／计数器的比较匹配 A 中断触发。若发生了定时器／计数器 1 中的比较匹配 A,则中断（在向量$ 004）被执行。在定时器／计数器中断标志寄存器——TIFR 中的定时器／计数器 1 中的比较 A 标志被设为 1。

位 5–OCIE1B：定时器/计数器 1 输出比较匹配 B 中断触发

当 OCIE1B 被设为 1，且状态寄存器中的 I 位被设为 1 时，定时器／计数器 1 的比较匹配 B 中断触发。若发生了定时器／计数器 1 中的比较匹配 B，则中断（在向量$005）被执行。在定时器／计数器中断标志寄存器——TIFR 中的定时器／计数器 1 中的比较 B 标志被设为 1。

位 4,2,0–Res：保留位

90 系列单片机的该位为保留位，总读 0。

位 3–TICE1：定时器/计数器 1 输入捕获中断触发

当 TICIE1 被设为 1，且状态寄存器中的 I 位被设为 1 时，定时器／计数器 1 输入捕获事件中断触发。若在引脚 31，即 PD6（ICP）上发生捕获触发事件，则中断（在向量$003）被执行。在定时器／计数器中断标志寄存器——TIFR 中的定时器／计数器 1 输入捕获标志被设置为 1。

位 1–TOIE0：定时器/计数器 0 溢出中断触发

当 TOIE0 被设为 0，且状态寄存器中的 I 位被设为 1 时，定时器／计数器 0 溢出中断触发。若在定时器／计数器 0 上发生溢出，则中断（在向量$008）被执行。在定时器／计数器的中断标志寄存器——TIFR 中的溢出标志（定时器 0）被设置为 1。

四、定时器／计数器中断标志寄存器——TIFR

位	7	6	5	4	3	2	1	0	
$38($58)	TOV1	OCF1A	OCIFB	-	ICF1	-	TOV0	-	TIFR
读／写	R／W	R／W	R/W	R	R/W	R	R/W	R	
初始化值	$00								

位 7–TOV1：定时器/计数器 1 溢出标志

当定时器／计数器 1 中产生溢出时，TOV1 位被设为 1。当执行相应中断处理向量时，TOV1 被硬件复位。TOV1 可由写入一个逻辑 1 到标志位而被清除。当 SREG（状态寄存器）的 I 位和 TOIE1（定时器／计数器 1 溢出中断触发），以及 TOV1 被设为 1 时，定时器／计数器 1 的溢出中断被执行。在 PWM 模式下，当定时器／计数器 1 在$0000 变化计数方向时，该位被设置为 1。

位 6–OCF1A：输出比较标志 1A

当 OCR1A,即输出比较寄存器 1A 中的数据与定时器／计数器 1 之间发生比较匹配时,OCF1A 被

设为 1。当执行相应中断处理向量时，OCF1A 由硬件清除。OCF1A 亦可通过向标志写逻辑 1 来清除。当 SREG 中的 I 位，OCIE1A（即定时器／计数器 1 比较匹配 A 中断触发），以及 OCF1A 被设为 1 时，定时器／计数器 1 比较匹配中断触发被执行。

位 5—OCF1B：输出比较标志 1B

当 OCR1B 即输出比较寄存器 1B 中的数据与定时器／计数器 1 之间发生比较匹配时，OCF1B 被设为 1。当执行相应中断处理向量时，OCF1B 由硬件清除。OCF1B 亦可通过向标志写逻辑 1 来清除。当 SREG 中的 I 位，OCIE1B(即定时器／计数器 1 比较匹配 B 中断触发)，以及 OCF1B 被设为 1 时，定时器／计数器 1 比较匹配中断触发被执行。

位 4，2，0——Res：保留位

90 系列单片机的该位为保留位，总读 0。

位 3——ICF1：中断捕获标志 1

当一个输入捕获事件发生时，ICF1 位标志被设为 1。表明定时器／计数器 1 的值已被输送到输入捕获寄存器——ICR1。当执行相应中断处理向量时，ICF1 被硬件清除。ICF1 亦可通过向标志写逻辑 1 来清除。

位 1——TOV0：定时器/计数器 0 位溢出标志位

当定时器／计数器 0 中产生溢出时，TOV0 位被设为 1，当执行相应中断处理向量时，TOV0 被硬件清除。TOV0 可由写入一个逻辑 1 到标志位而被清除。当 SREG 的 I 位和 TOIE0（定时器／计数器 0 溢出中断触发），以及 TOV0 被设为 1 时，定时器／计数器 0 的溢出中断被执行。

2.5.2 外部中断

外部中断由引脚 INT1 和 INT0 触发。请注意，在触发时，即使 INT0 和 INT1 引脚被设置为输出，亦会触发中断。这一特征可用来进行软件中断。外部中断可通过上升或下降边沿及低电平来触发。这在 MCU 控制寄存器——MCUCR 中已说明。当外部中断触发，且被设置为电平触发时，只要引脚保持低电平，中断将被触发。

外部中断的设置已在 MCU 控制寄存器——MCUCR 的规格中被说明。

2．5．3 中断应答时间

AVR 所有允许的中断执行应答最少为 4 个时钟周期。在 4 个时钟周期之后，用于现行中断控制程序的程序向量寻址被执行。在 4 个时钟周期中，程序计数器（2 个字节）被压入堆栈，且堆栈指针被减量 2。该向量为一个向中断程序的相关转移指令，需 2 个时钟周期。若在一个多周期指令的执行中发生中断，该指令在中断执行前结束。

从中断处理程序（像调用子程序）的返回需要 4 个时钟周期。在这 4 个时钟周期之中，程序计数器（2 个字节）从堆栈中被弹出，且堆栈指针被增量 2。当 AVR 从某一中断中出来时，它总是返回到主程序，并在任何挂起的中断执行前执行多于一条的指令。

注意：状态寄存器——SREG 不由 AVR 硬件处理，也不为中断或子程序工作。由于中断处理程序需要一个 SREG 的存储。这一切均由用户软件完成。

由事件发生的中断触发能保留状态，即当事件发生时，中断标志被设置。若中断标志已被清除，但中断条件持续，则在下次事件发生时，中断标志才被设置。

2．5．4　MCU 控制寄存器 MCUCR

MCUCR 控制寄存器包含通用 MCU 功能的控制位。

位	7	6	5	4	3	2	1	0
$35（$55	SRE	SRW	SE	SM	ISC11	ISC10	ISO01	ISC00	MCUCR
读／写：	R／W							
初始化值：	$00							

位 7——SRE：外部 SRAM 触发

当 SRE 被设置为 1 时，外部的 SRAM 被触发，且引脚 AD0~AD7（A 口），A8~A15（C 口），RD 和 WR（D 口）作为第二功能被触发。 SRE 位覆盖任何的数据方向寄存器的方向设置，详见“2．3．2 节内部和外部的 SRAM 数据存储器”，该部分描述了外部 SRAM 的引脚功能。当 SRE 位被清除时，外部数据 SRAM 被禁止，作为通常的引脚且用于数据方向的设置。

位 6--SRW：外部 SRWM 等待状态

当 SRW 设置为 1 时，在外部的数据 SRAM 访问周期中插入一个周期的等待状态。当 SRW 被清 0 时，外部数据 SRAM 的访问使用正常的 2 周期。详见图 2．12 无等待状态的外部数据 SRAM 访问周期和图 2.1 有等待状态的外部数据 SRAM 访问周期。

位 5--SE: 休眠触发

SE 位须设为 1，以便当 SLEEP 指令执行时，使 MCU 进入休眠模式。除非出于编程的目的，为防止 MCU 进入休眠模式，建议用户只在执行 SLEEP 指令之前才设置休眠触发 SE 位。

位 4 --SM: 休眠模式

这一位可选择两种休眠模式。当 SM 被清为 0 时，闲置模式被选为休眠模式。当 SE 设为 1 时，掉电模式被选为休眠模式。请参考“2．6．1 节休眠状态”。

位 3,2——ISC11，ISC10：中断检测控制 1，位 1 和位 0

如果 SREG 的 I 标志位和 GIMSK 寄存器中的相应中断屏蔽已被设置，则外部中断 1 被外部引脚 INT1 触发。由表 2．4 定义触发中断的外部 INT1 引脚上的是电平还是边沿。

位 1，0——ISC01，ISC00：中断检测控制 0，位 1 和位 0

如果 SREG 的 I 标志位和 GIMSK 寄存器中的相应中断屏蔽已被设置，则外部中断 0 被外部引脚 INT0 触发。由表 2．5 定义触发中断的外部 INT0 引脚上的是电平还是边沿。

表 2.4　中断 1 检测控制

ISC11	ISC10	说　明
0	0	INT1 的低电平产生一个中断请求
0	1	INT1 的高电平产生一个中断请求
1	0	INT1 的下降沿产生一个中断请求
1	1	INT1 的上升沿产生一个中断请求

表 2.5　中断 0 检测控制

ISC01	ISC00	说　明
0	0	INT0 的低电平产生一个中断请求
0	1	INT0 的高电平产生一个中断请求
1	0	INT0 的下降沿产生一个中断请求
1	1	INT0 的上升沿产生一个中断请求

注意：当改变 ISC10／ISC00 位时，INT0 须通过清除其在 GIMSK 寄存器中的中断触发位来加以禁止，否则当该位改变时，一个中断就会产生。

2．6　AVR 单片机的节电方式

2．6．1　休眠状态

为了进人休眠状态，MCUCR 中的 SE 位被设为 1，且须执行一条 SLEEP 指令。当 MCU 在休眠模式下发生了一个允许的中断，MCU 唤醒，执行中断程序，并恢复 SLEEP 后面指令的执行。寄存器堆的内容和 I／O 存储器没有改变。若在休眠模式下发生复位，MCU 被唤醒，并从复位向量执行。

注意：若为了从掉电状态下唤醒 MCU 而使用了某一电平触发的中断，必须保持 16 ms 的低电平，即长于晶振启动的时间；否则，在 MCU 开始执行以前，中断标志位有可能返回零值。

2.6.2　闲置模式

当 SM 位被清为 0，SLEEP 指令使 MCU 进入闲置状态，这时 CPU 停止工作，而定时器／计数器、看门狗以及中断系统继续工作。这使得 CPU 可以由外部触发的中断来唤醒，亦可由内部诸如定时器溢出中断和看门狗复位来唤醒，不需要通过模拟比较器中断来唤醒 CPU，这样可通过设置模拟比较器的控制状态寄存器——ACSR 中的 ACD 位来对模拟比较器进行掉电。因此在闲置状态下减少了功耗。

2.6.3　掉电模式

当 SM 位被设为 1 时，SLEEP 指令使 MCU 进入掉电状态。在此模式下，外部晶振停止工作。用户可在掉电模式下选择看门狗是否触发。若看门狗为触发，当看门狗超过超时规定的时间时，它会唤醒 MCU。若看门狗为非触发，只有外部的复位或外部电平触发中断可唤醒 MCU。

2.7　AVR 单片机定时器／计数器

AT90S8515 单片机有 2 种通用定时器／计数器，即一个 8 位的定时器／计数器、一个 16 位的定时器／计数器。定时器／计数器从 10 位的预定比例定时器获取独立的预定比例区域。2 个定时器／计数器可被用作带内部时钟的时基定时器，或被用作带可触发计数的外部引脚连接的计数器。

2.7.1　定时器／计数器预定比例器

图 2．20 所示为通用定时器／计数器的预定比例器。

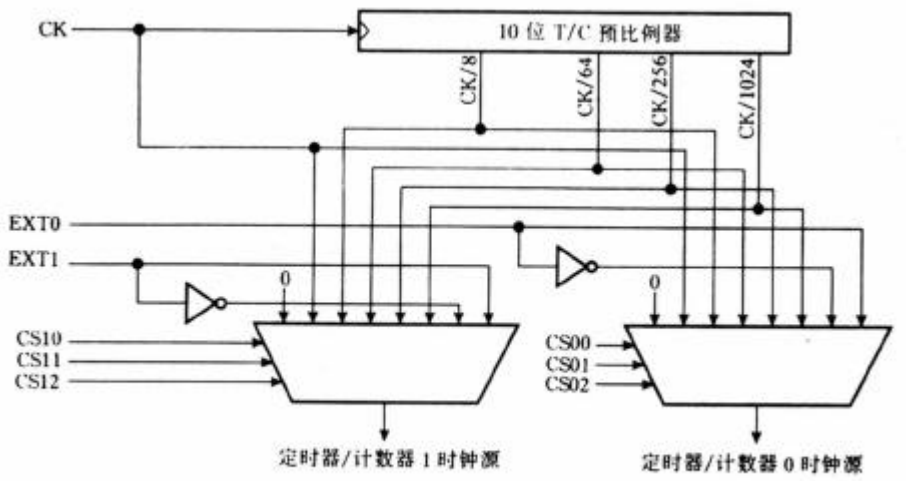


图 2.20　定时器/计数器预定比例器

四个不同的预定比例部分为：CK／8、CK／64、CK／256 和 CK／1024，图中 CK 为晶振时钟。对于两个定时器／计数器，新加的部分为 CK。外部源和停止均可选为时钟源。

2.7.2　8位定时器／计数器 0

图 2．21 所示为定时器／计数器 0 的方框图。

8 位的定时器／计数器 0 可从 CK、预定比例器时钟源，或外部引脚来选择时钟源。另外，它还可以像走时器／计数器 0 的控制寄存器——TCCR0 格式中所描述的那样而停止。

溢出状态比例标志位在定时器／计数器中断比例标志位寄存器——TIFR 中。定时器／计数器 0 的中断触发／禁止设置在定时器／计数器中断的控制屏蔽寄存器——TIMSK 中。

当定时器／计数器 0 被外部定时时，外部信号与 CPU 的振荡频率同步。为了确保外部时钟获取正确的采样，在两个外部时钟转换之间的最少时间必须维持一个内部 CPU 的时钟周期。外部时钟信号在内部 CPU 时钟的上升边沿被采样。

8 位的定时器／计数器 0 有机会用于更低的预定比例时，则具有高分辨率和高准确性的特性。类似地，高预定比例特性使得定时器／计数器 0 对低速功能，或不经常操作的精确定时功能很有用。

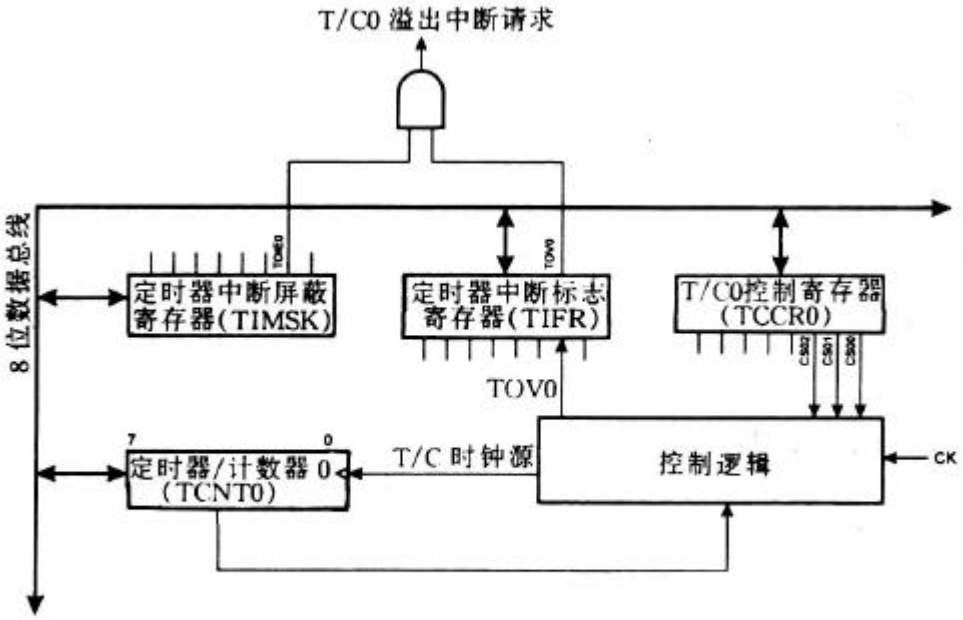


图 2.21　定时器/计数器 0 方框图

一、定时器／计数器 0 的控制寄存器——TCCRO

位	7	6	5	4	3	2	1	0
$33（$53	--	--	--	--	--	CS02	CS01	CS00	TCCR0
读／写:	R	R	R	R	R	R／W	R/W	R/W	
初始化值:	$00								

位 7~3—Res：保留位

90 系列单片机的这些位为保留位，总读 0。

位 2,1,0——CS02,CS01,CS00:时钟选择 0 的位 2、1 和 0

时钟选择 0 的 2、1 和 0 位定义定时器 0 的预定比例源，见表 2．6。

停止条件提供定时器触发／禁止功能。时钟分频的模式从晶振时钟直接换算。若使用外部引脚模式，相应设置必须在实际数据方向控制寄存器中完成。

表 2.6　时钟 0 预定选择

CS02	CS01	CS00	说　明	CS02	CS01	CS00	说　明
0	0	0	停止，定时器/计数器 0 被停止	1	0	0	CK/256
0	0	1	CK	1	0	1	CK/1 024
0	1	0	CK/8	1	1	0	外部 T0 脚，下降沿
0	1	1	CK/64	1	1	1	外部 T0 脚，上升沿

表 2.6　时钟 0 预定选择

二、定时器计数器 0——TCNT0

位	7	6	5	4	3	2	1	0
$32（$52）	MSB						LSB	TCNT0
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始化值:	$00							

定时器／计数器 0 是带读写访问的向上计数器。若定时器／计数器 0 被写入，同时时钟源正被执行，定时器／计数器 0 在写人操作之后继续计数。

2.7.3　16 位定时器／计数器 1

图 2.22 所示为定时器／计数器 1 的方框图。

16 位的定时器／计数器 1 可以从 CK、预定比例器时钟源或外部引脚中选择时钟源。另外，它还可以像在定时器／计数器 1 控制寄存器——TCCR1A、TCCR1B 中描述的那样被停止。在定时器／计数器控制寄存器——TCCR1A、TCCR1B 中可以找到不同的状态标志（溢出、比较匹配以及捕获事件）和控制信号。对定时器／计数器 1 的中断触发／禁止设置可以在定时器／计数器中断屏蔽寄存器——TIMSK 中找到。

当定时器／计数器 1 被外部定时时，外部信号与 CPU 振荡频率同步。为确保从外部时钟获取正确的采样，在 2 个外部时钟转换之间的最小时间至少为一个内部 CPU 时钟周期。外部时钟信号在内部 CPU 时钟的上升边沿被采样。

16 位的定时器／计数器 1 有机会用于更低的预定比例时，则具有高分辨率和高准确性的特性。类似地，高预定比例特性使得定时器／计数器 1 对低速功能，或不经常操作的精确定时功能很有用。

定时器／计数器 1 提供输出比较寄存器 1A 和 1B——OCR1A 和 OCR1B 的两个输出比较功能作为与定时器／计数器 1 内容进行比较的数据源。输出比较功能包括比较 A 匹配的计数器可选择的清除，以及在比较匹配输出比较引脚上的操作。

定时器／计数器可被用作 8 位、9 位或 10 位脉冲调制器。在此模式下，定时器和 OCR1A／OCR1B 寄存器具有集中脉冲双抗误操作独立的 PWM 能力。

定时器／计数器的输入捕获功能提供了对定时器／计数器 1 向输入捕获寄存器——ICR1 内容的捕获，该捕获操作由在输入捕获引脚——ICP 上的外部事件触发。实际的捕获事件设置由定时器／计数器 1 的控制寄存器——TCCR1B 来定义。另外，模拟比较器可触发该输入捕获。请参照“模拟比较器”部分。ICP 的引脚逻辑如图 2．23 所示。定时／计数器 1 的输入捕获噪声消除器示意图见图 2．24。

如果噪音清除器为触发，则捕获事件的实际触发条件在捕获被触发前受到多于 4 个采样的监测。输入引脚信号以 XTAL 的时钟频率被采样。

一、定时器／计数器 1 控制寄存器 A——TCCR1A

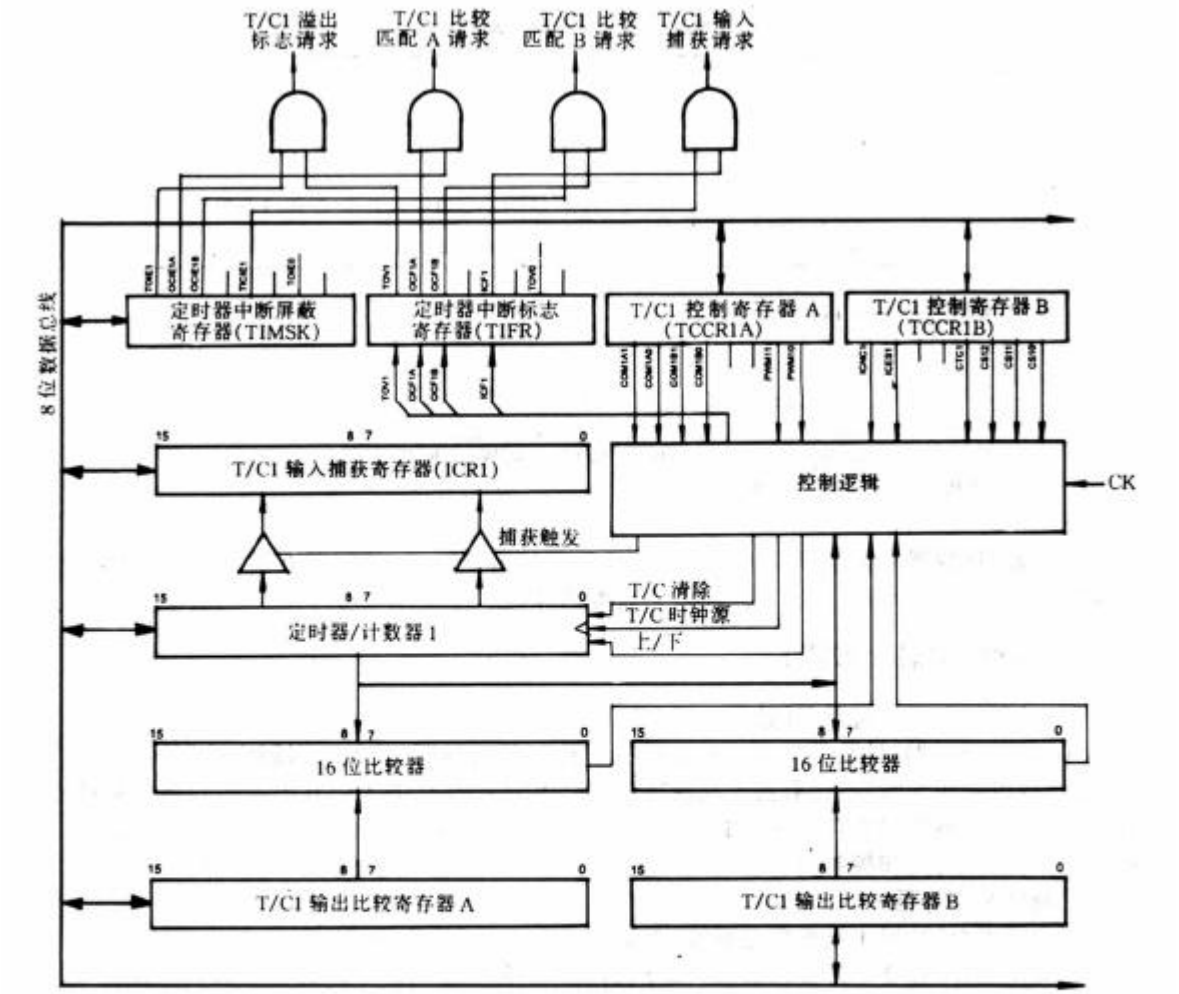


图 2.22　定时器/计数器 1 方框图

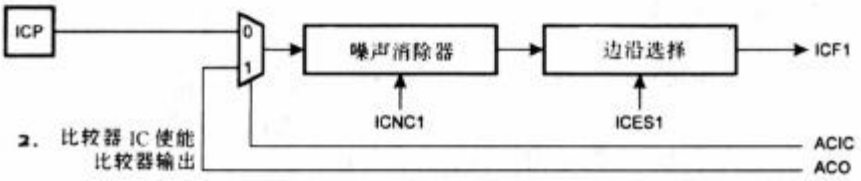


图 2.23　ICP 的引脚原理图

位	7	6	5	4	3	2	1	0
$2F（$4F）	COM1A1	COM1A0	COM1B1	COM1B0	--	--	PWM11	PWM10	TCCR1A
读／写:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
初始化值:	$00								

位 7,6——COMM1A1，COM1A0: 比较输出模式 1，位 1 和 0

COM1A1 和 COM1A0 控制位决定了在定时器／计数器 1 中比较匹配之后的输出引脚事件。输出引脚事件影响 OC1A，即输出比较 A 引脚 1。由于这是对 I／O 口的可替换功能，相应的方向控制位必须设为 1，以便对输出引脚进行控制。控制设置如表 2．7 所示。

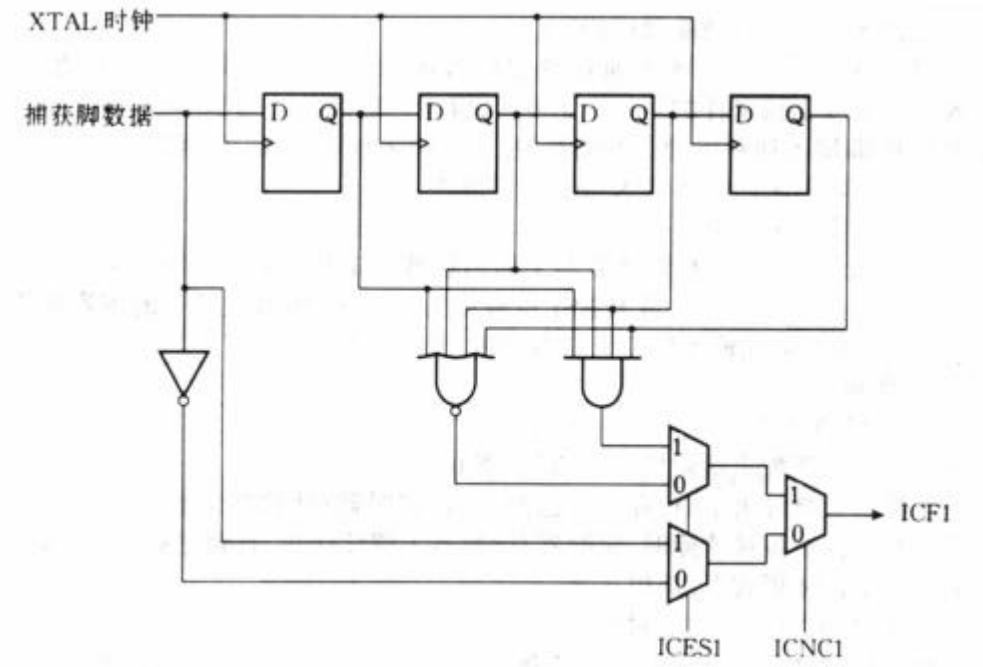


图 2.24　输入捕获噪声消除器示意图

位 5,4——COMM1B1，COM1B0: 比较输出模式 1，位 1 和 0

COM1B1 和 COM1B0 控制位决定了在定时器/计数器 1 中比较匹配之后的输出引脚事件。输出引脚事件影响 OC1B，即输出比较 B 引脚 1。由于这是对 I／O 口的可替换功能，相应的方向控制位必须设为 1，以便对输出引脚进行控制。控制设置如表 2．7 所示。

在 PWM 模式下，这些位有不同的功能，请参考表 2．11 的具体说明。

当变换 COM1X1 和 COM1X0 位时，输出比较器中断 1 必须通过清除 TIMSK 寄存器中的中断触发位来禁止；否则在位变换时，会发生中断。

位 3,2——Res:保留位

90 系列单片机的该位为保留位，总读 0。

位 1,0--PWM11,PWM10: 脉冲宽度调制器选择位

这些位如表 2．8 中指明的一样选择定时器／计数器 1 的 PWM 操作。

表 2.7　比较 1 模式选择

COM1X1	COM1X0	说　明
0	0	定时器/计数器 1 与输出脚 OC1X 不连接
0	1	触发 OC1X 输出线
1	0	清除 OC1X 输出线(为 0)
1	1	设置 OC1X 输出线(为 1)

X=A 或 B

表 2.8　PWM 模式选择

PWM11	PWM10	说　明
0	0	禁止定时器/计数器 1 的 PWM 操作
0	1	定时器/计数器 1 为 8 位 PWM
1	0	定时器/计数器 1 为 9 位 PWM
1	1	定时器/计数器 1 为 10 位 PWM

二、定时器／计数器 1 控制寄存器 B——TCCR1B

位	7	6	5	4	3	2	1	0
$2E（$4E）	ICNC1	ICES1	--	--	CTC1	CS12	CS11	CS10	TCCR1B
读／写:	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
初始化值:	$00								

位 7——ICNC1：输入捕获噪音清除器（4CKS)

当 ICNC1 位被清为 0 时，输入捕获触发噪音清除器功能被禁止。输入捕获在指定的 ICP，即输入捕获引脚上被采样的第一个上升/下降沿处被触发。当 ICNC1 被设为 1，4 个延续的采样成为 ICP，即输入捕获引脚上的测量值，所有的采样须为高/低，取决于 ICES1 位的输入捕获触发特性。实际的采样频率为 XTAL 时钟频率。

位 6——ICES1：输入捕获 1 边沿选择

当 ICES1 位被清为 0，定时器/计数器 1 的内容被传输到输入捕获寄存器——ICR1，即在输入捕获引脚 ICP 的下降边沿。当 ICES1 位被设为 1，定时器/计数器 1 的内容被传输到输入捕获寄存器——ICR1，即在输入捕获引脚 ICP 的上升边沿。

位 5,4——Res:保留位

90 系列单片机的该位为保留位，总读 0。

位 3——CTC1：在比较匹配上清除定时器/计数器 0

当 CTC1 控制位被设为 1 时，在比较匹配之后，定时器/计数器 1 被子复位到时钟周期中的$0000。若 CTC1 控制位被清除时，定时器/计数器 1 继续计数，直到它被停止、清除、溢出，或被改变方向。在 PWM 模式下，该位无效。

位 2,1,0——CS12,CS11,CS10:时钟选择 1 的位 2、1 和 0

时钟选择 1 的位 2、1 和定义了定时器/计数器 1 的预定比例源，见表 2.9

表 2.9　时钟预定比例选择

CS02	CS01	CS00	说　明	CS12	CS11	CS10	说　明
0	0	0	停止，定时器/计数器 1 被停止	1	0	0	CK/256
0	0	1	CK	1	0	1	CK/1 024
0	1	0	CK/8	1	1	0	外部 T1 脚，下降沿
0	1	1	CK/64	1	1	1	外部 T1 脚，上升沿

三、定时器／计数器 1——TCNT1H 和 TCNT1L

位	15	14	13	12	11	10	9	8	
$2D（$4D）	MSB		--	--					TCNT1H
$2C ($4C)								LSB	TCNT1L
	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	
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值:	$00 00								

这个 16 位的寄存器包括 16 位定时器/计数器 1 的预定比例值.为确保当 CPU 访问这些寄存器时,高低字节被同时读写,使用一个 8 位的暂存寄存器(TEMP)来完成访问.

1.TCNT1 定时器/计数器 1 写入

当 CPU 向高位字节 TCNT1H 写入时,写入的数据被放入 TEMP 寄存器中,然后,当 CPU 向低位字节 TCNT1L 写入时,数据的字节被与 TEMP 寄存器中的字节数据组合,且全部的 16 位被同步地向

TCNT1 定时器/计数器 1 寄存器写入。作为结果,高字节的 TCNT1H 必须被先访问，以便完成全 16 位寄存器的写人操作。

2．TCNT1 定时器／计数器 1 读取

当 CPU 读低位字节 TCNT1L 时，TCNT1L 低字节的数据被送到 CPU，且高字节 TCNTlH 的数据被放置于 TEMP 寄存器中。当 CPU 读高位字节 TCNT1L 时，CPU 接收 TEMP 寄存器中的数据。作为结果，低字节的 TCNT1L 必须先被访问，以便完成全 16 位寄存器的读取操作。定时器／计数器 1 随着读和写访问，实行向上计数或向上／向下计数（在 PWM 方式）。如果定时器／计数器已被写入且时钟源已被选择，则定时器计数器 1 在被设置后的定时时钟周期内连续计数。

四、定时器／计数器 1 输出比较寄存器——OCR1AH 和 OCR1AL

位	15	14	13	12	11	10	9	8	
$2B（$4B）	MSB		--	--					OCR1AH
$2A ($4A)								LSB	OCR1AL
	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	
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值:	$00 00								

五、定时器／计数器 1 输出比较寄存器——OCR1BH 和 OCR1BL

位	15	14	13	12	11	10	9	8	
$29（$49）	MSB		--	--					OCR1BH
$28 ($48)								LSB	OCR1BL
	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	
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值:	$00 00								

输出比较器为一个 16 位的读／写寄存器。定时器／计数器 1 输出比较寄存器包括了将要连续地与定时器计数器 1 相比较的数据。比较匹配的操作在定时器／计数器 1 的控制和状态寄存器中被区分。

由于输出比较寄存器——OCR1A 和 OCR1B 为一个 16 位的寄存器，当 OCR1A/B 被写入时，须使用临时寄存器 TEMP 来确保全部的字节被同时写入。当 CPU 写高位字节时，OCR1AH 或 OCR1BH 数据被临时地存储在寄存器 TEMP 中。当 CPU 写低位字节时，OCR1AL 或 OCR1BL、TEMP 寄存器被同步地向 OCR1AH 或 OCR1BH 写入。作为结果，高位的 OCR1AH 或 OCR1BH 必须被先写入，以便完成全部的 16 位寄存器写入操作。

六、定时器／计数器 1 输入捕获寄存器——ICR1H 和 ICR1L

位	15	14	13	12	11	10	9	8	
$25（$45）	MSB								ICR1H
$24 ($44)								LSB	ICR1L
	7	6	5	4	3	2	1	0	
读／写:	R	R	R	R	R	R	R	R	
读／写:	R	R	R	R	R	R	R	R	
初始化值:	$00 00								

输入捕获寄存器为一个 16 位的只读寄存器。当在输入捕获引脚 ICP 上信号的上升或下边沿（根据输入捕获边沿设置——ICES1)被检测到时，定时器／计数器 1 的当前值被传输到输入捕获寄存器——ICR1。同时，输入捕获标志——ICF1 被设为 1。

由于输入捕获寄存器——ICR1 为一个 16 位的寄存器，当 ICR1 被读出时，使用了一个临时寄存器 TEMP 以便确保全部的字节被同时读出。当 CPU 读取低位字节 ICR1L 时，数据被送入 CPU 且高位字节 ICR1H 的数据被放置在 TEMP 寄存器中。当 CPU 读取高位字节 ICR1H 中的数据时，CPU 接收 TMEP 寄存器中的数据。作为结果，低位字节 ICR1L 必须先被访问到，以便完成一个全 16 位寄存器的读取操作。

七、PWM 模式下的定时器／计数器 1

当选择 PWM 模式时，定时器／计数器 1 以及输出比较寄存器 OCR1A 和输出比较寄存器 OCR1B 形成一个双 8 位、9 位或 10 位的自运行，抗误操作，且在引脚 PD5（OC1A）和 OC1B 引脚上带有输出的节拍修正 PWM。定时器／计数器 1 作为向上／向下的计数器，从$ 0000 向上计数到 TOP（参考表 2．10），在重复循环之前，它反转，并向下再次计数到 0。

当计数器值与 OCR1A、OCR1B 的最后 10 位相配时，PD5（OC1A）／OC1B 引脚根据在定时器／计数器 1 控制寄存器 TCCR1A 中 COM1A1／COM1A0 或 COM1B1／COM1B0 位的设置而被设置或被清除，请参考表 2．11。

表 2.10　定时器 TOP 值和 PWM 频率

PWM 分辨率	定时器 TOP 值	频　率
8 位	$ 00FF(255)	$f_{TCI}/510$
9 位	$ 01FF(511)	$f_{TCI}/1022$
10 位	$ 03FF(1 023)	$f_{TCI}/2046$

表 2.11　在 PWM 方式时比较 1 方式选择

COM1X1	COM1X0	在 OCX1 上的作用
0	0	不连接
0	1	不连接
1	0	清比较匹配，向上计数，置比较匹配，向下计数(PWM 不翻转)
1	1	清比较匹配，向下计数，置比较匹配，向上计数(PWM 翻转)

X＝A or B

注意：在 PWM 模式下，当后 10 位 OCR1A／OCR1B 位被写入时，它们被送入临时地址。当定时器／计数器 1 到达 TOP 时，它们被锁存。这就防止了在非同步 OCR1A／OCR1B 写入事件中发生奇数长的 PWM 脉冲（误操作）。见图 2．25 的例子。

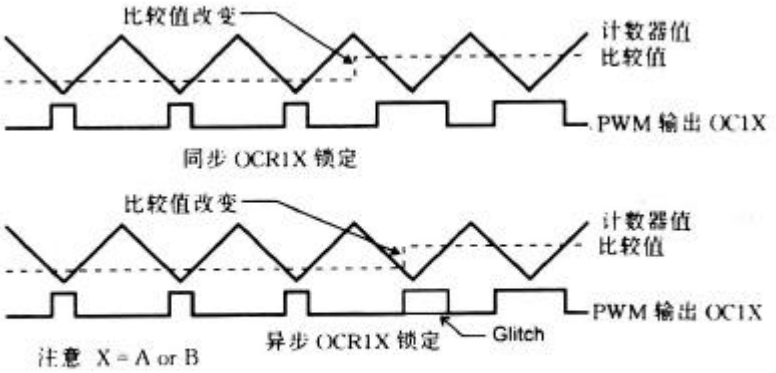


图 2.25　有效的非同步 OCR1 锁存

当 OCR1A 包含$ 0000 或 TOP 输出　OC1A／OC1B 根据 COM1A1 和 COM1A0 或 COM1B1／COM1B0 的设置保持低或高，如表 2.12 所示。

在 PWM 模式下，定时器溢出标志 1、TOVI，当计数器在方向$ 0000 时被设置。定时器溢出中断 1 以正常的定时器／计数器模式工作，比如，当 TOV1 被设置，从而提供了定时器溢出中断 1 和全局中断为触发时，它被执行。这也同样用于定时器输出比较 1 的标志和中断。

表 2.12　PWM 输出 OCRIX 好于 $0000 成 TOP

COM1X1	COM1X0	OCR1X	OC1X 输出
1	0	$0000	L
1	0	TOP	H
1	1	$0000	H
1	1	TOP	L

2.7.4　看门狗定时器

一、看门狗定时器

看门狗定时器由片内一个独立的 1MHz 分开的晶振定时。通过控制看门狗定时器的预定比例器，看门狗的复位间歇从 16 周期到 2048 周期之间调整。这些值适应 Vcc一5V。请参考其它 Vcc 电压下的 RC 晶振频率的特性化数据。WDR，即看门狗复位指令对看门狗定时器进行复行位。有 8 种不同的时钟周期可供选择，来决定在 2 个 WDR 指令之间的最大时间，这样做是为了避免看门狗定时器对 MCU 进行复位。若复位周期结束而无其它的 WDR 指令，90 系列单片机进行复位，并从复位向量执行。参见图 2.26。

为了防止意外禁止看门狗定时器，当看门狗被禁止时必须服从一个特定的关断顺序，请详见看门狗的控制寄存器。

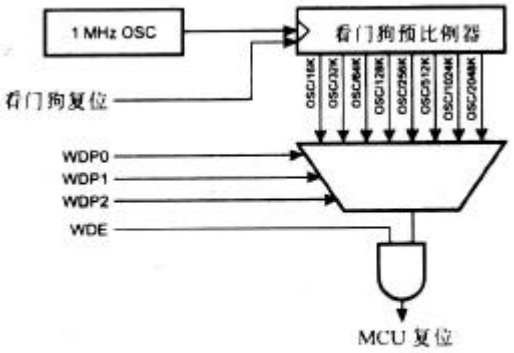


图 2.26　看门狗定时器

二、看门狗定时器控制寄存器——WDTCR

位	7	6	5	4	3	2	1	0
$21（$41）			--	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
读／写:	R	R	R	R/W	R/W	R/W	R/W	R/W	

初始化值:　$00

位 7～5—Res: 保留位

90 系列单片机的这些位为保留位，总读 0。

位 4—WDTOE：看门狗关断触发

当 WDTOE 被清除时该位必须被置 1，否则看门狗将不会被禁止，一旦置位后，硬件将在 4 个时钟周期后清除该位。请参照看门拘禁止过程的 WDE 位的描述。

位 3—WDE：看门狗触发

当 WDE 时改设为 1 时，看门狗定时器触发。若 WDE 被清为 0，看门狗定时器功能被禁止。WDE 仅在 WDTOE 位设置时被清除，为了禁止被触发的看门狗定时器，必须遵守以下过程：

（1）在同一个操作中，把 WDTOE 和 WDE 写成 1，即使在禁止操作开始前 WDE 为 1，也必须把 1 写入 WDE。

（2）在随后 4 个机器周期中，把 WDE 写为 0，这会禁止看门狗。

位 2～0—WDP2～0：看门狗定时器预定比例器 2、1、0

WDP2、WDP1、WDP0 决定了当看门狗定时器触发时，看门狗定时器的预定比例。不同的预定比例值以及它们相应的超时时间，如表 2．13 所示。

表 2.13 看门狗定时器预定比例选择(典型值 V_{CC} = 5V)

WDP2	WDP1	WDP0	定时器输出周期/ms	WDP2	WDP1	WDP0	定时器输出周期/ms
0	0	0	16	1	0	0	256
0	0	1	32	1	0	1	512
0	1	0	64	1	1	0	1 024
0	1	1	128	1	1	1	2 048

2.8　AVR 单片机　E2PROM 读 / 写访问

I / 0 空间中可以访问 E2PROM 寄存器。

写入访问时间在 2..5~4 ms 之间。取决于 Vcc 电压。自定时功能使得用户软件检测下一个字节何时被写入。若 Vcc 低于一定的电平，E2PROM 降低电压被检测，防止了对 E2PROM 的写入。当 E2PROM 被读取或写入时，在下一个指令执行前，CPU 被中止达两个时钟周期。

一、E2PROM 地址寄存器——EEAR

位	15	14	13	12	11	10	9	8
$1F（$3F）	--	--	--	--	--	--	--	EEAR9	EEARH
$1E ($3E)	EEAR7	EAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0
读 / 写:	R	R	R	R	R	R	R	R/W
读 / 写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始化值:	$00 00							

E2PROM 地址寄存器——EEARH 和 EEARL，指定了在 512 字节的 E2PROM 空间中的 E2PROM 地址。E2PROM 数据字节被在 0~ 511 之间线性地编址。

二、E2PROM 数据寄存器——EEDR

位	7	6	5	4	3	2	1	0
$1D（$3D）	MSB						LSB	EEDR
读 / 写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始化值:	$00							

位 7~0——EEDR7~0：E2PROM 数据

对于 E2PROM 写入操作，EEDR 寄存器包含了写入 E2PROM 的数据，由 EEAR 寄存器给出其地址。对于 E2PROM 的读取操作，EEDR 包含由 EEAR 给出的 E2PROM 地址，数据将从这一地址中读出。

三、E2PROM 控制寄存器——EECR

位	7	6	5	4	3	2	1	0
$1C（$3C）	--	--	--	--	--	EEMWE	EEWE	EERE	EECR
读 / 写:	R	R	R	R	R	R/W	R/W	R/W
初始化值:	$00							

位 7~3——Res:保留位

90 系列单片机的这些位为保留位，总读 0。

位 2——EEMWE：E2PROM 的主写触发

EEMWE 位决定了设置 EEWE 是否导致 EERPOM 被写入。当 EEMWE 被设为 1 时，设置 EEWE 将把数据写入 E2PROM 所选择的地址中。如果 EEMWE 为 0，则设置 EEWE 无效。当 EEMWE 被软件设置后，4 个周期后被硬件清除。详见 E2PROM 写过程中的 EEWE 位的描述。

位 1——EEWE：E2PROM 写入触发

E2PROM 写入触发信号 EEWE 是对 E2PROM 的写入选通。若地址和数据被正确设置，EEWE 位必须被设置从而写 E2PROM。当 EEWE 被置 1 时，EEMWE 必须被置 1，否则不会发生 E2PROM

的写操作。当写入 EERPOM 时应服从以下的过程（第（2）步和第（3）步不是必须的）：

(1）等待 EEWE 位变为 0；

（2）把新的 E2PROM 地址写入 EEAR（可选）；

（3）把新的 E2PROM 数据写入 EEDR（可选）；

（4）在 EECR 中的 EEMWE 位写逻辑 1；

（5）在设置 EEMWE 后的 4 个时钟周期内，在 EEWE 中写入逻辑 1。

在写人访问时间（一般为 5 V时 2.5 ms，2.7 V时 4 ms）过后， EEWE 由硬件清零。用户软件在写入下一个字节之前，查询这一位，并等待零值。当 EEWE 被设过后，CPU 在执行下一个指令前中止 2 个周期。

位 0——EERE：E2PROM 读取触发

E2PROM 读取触发信号 EERE 是对 E2PROM 的读取选通。当 EEAR 寄存器中的地址被正确设置时，EERE 必须被设置。当 EEAR 被硬件清空时，在 EEDR 寄存器中可找到所需数据。E2PROM 读取访问占用 1 个指令，无需查询 EERE 位。一旦 EERE 被设过后，CPU 在执行下一个指令前中止 2 个周期。在开始读操作之前，用户可以查询 EEWE 位。如果当新的数据或地址被写到 E2PROMI／O 寄存器时，一个写入操作在进行，则写入操作将被中断，并且结果是定义的。

2．9 AVR 单片机串行接口

2．9．1 同步串行接口 SPI

同步串行接口（SPI）允许在 90 系列单片机和外设或几个 90 系列单片机之间高速同步数据传送，见图 2．27。90 系列单片机 SPI 的特征如下：

（1）全双工，3 线同步数据传送。
（2）主从操作。
（3）5Mb/S 的位传送频率（最大值）。
（4）LSB 或 MSB 在先。
（5）四种可编程的位速率。
（6）传送停止的中断标志。
(7）写冲突标志保护。
（8）从闲置模式下唤醒（仅从模式）。

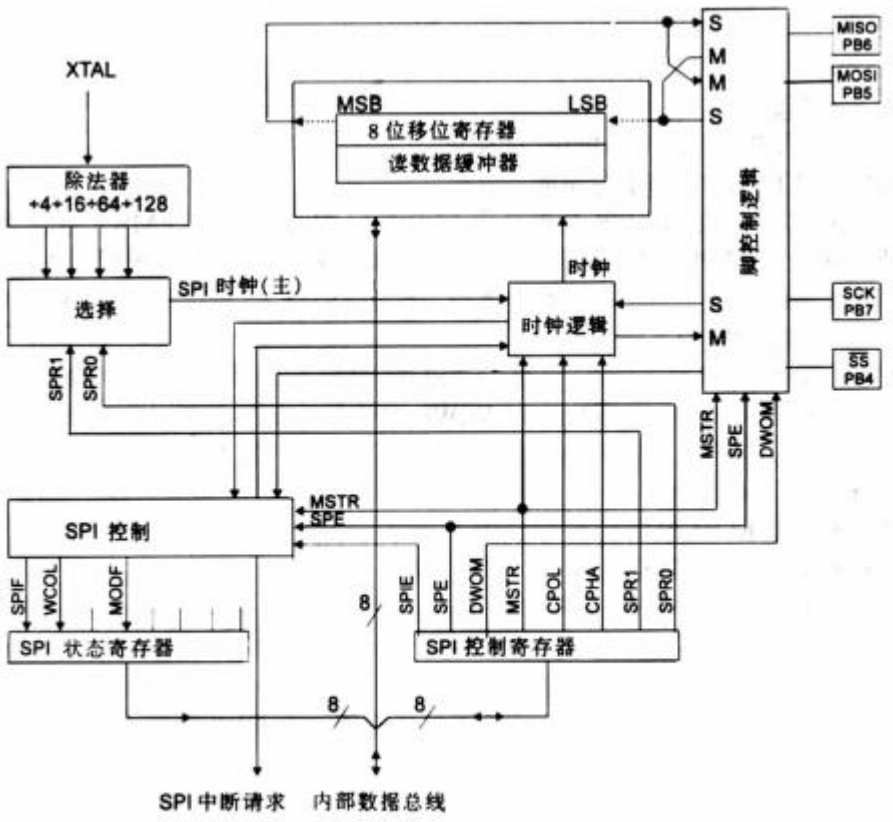


图 2.27　SPI 方框图

主从 CPU 之间的 SPI 连接如图 2．28 所示，PB7（SCK）引脚是主机模式的时钟输出和从机模式的时钟输入，把数据写入主 CPU 的 SPI 数据寄存器会启动 SPI 的时钟发生器，而数据从 PBS（MOSI）引脚移出和移入。在一个字节移出后，SPI 时钟发成器停止，设置传送停止标志位（SPIF）。如果 SPCR 寄存器中的中断触发位（SPIE）被设置，则生成一个中断请求。从机选择输入 PB4（/SS），

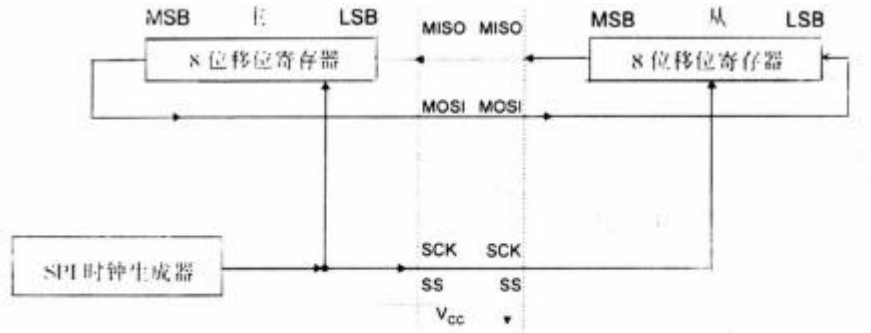


图 2.28　SPI 主从 CPU 内部连接

被设置为低来选择单独的 SPI 器件作为从机，主机和从机的两个移位寄存器可以被认为是一个分开的 16 位环形移位寄存器，如图 2，28 所示。当数据从主机移向从机，同时数据也移向相反的方向。这意味着在一个移位周期内，主机和从机的数据交换。

这个系统在发送方向上有一级缓冲而在接收方向有两级缓冲。这意味在全部的移位周期完成之前要被传送的字符不能被写入 SPI 数据寄存器。在接收数据时，在下一个字符被完全移入之前已经收到的数据必须从 SPI 数据寄存器中读走，否则，这个字符会丢失。

当 SPI 被触发时，MOSI、MISO、SCK 和/SS 引脚的数据方向，按照表 2. 14 中来配置。

表 2.14　SPI 脚配置

引脚	方向，主 SPI	方向，从 SPI	引脚	方向，主 SPI	方向，从 SPI
MOSI	用户定义	输入	SCIK	用户定义	输入
MISO	输入	用户定义	$\overline{SS}$	用户定义	输入

一、/SS 引脚的功能

当 SPI 被配置为主机时（SPCR 的 MSTR 置 1），用户可以决定/SS 引脚的方向。如果/SS 引脚被设为输出，该引脚作为通用输出不影响 SPI 系统；如果需被设为输入，则必须保持为高来保证主机 SPI 的操作。如果在主机模式下，/SS 引脚为输入，而且被外设电路置低，则该系统认为另外的主机选择该 SPI 为它的从机并开始对它传递数据。为了防止总线相连，SPI 系统将采取以下动作：

（1）SPCR 的 MSTR 位被清除而且 SPI 系统变成从机，结果是 MOSI 和 SCK 引脚变成输入。

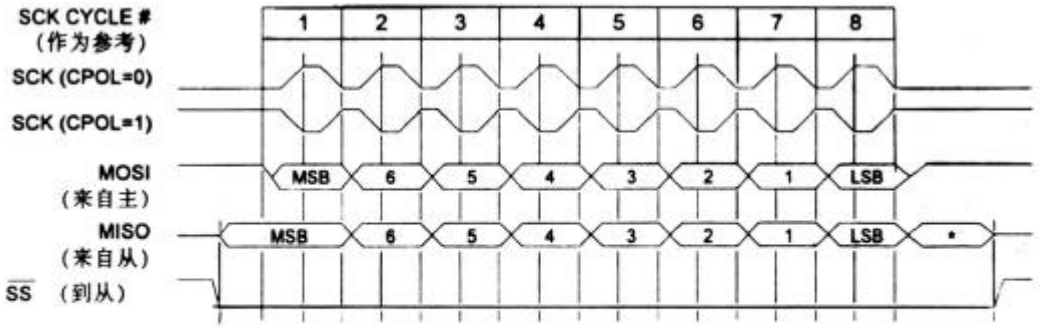
（2）SPSR 中的 SPIF 位被设置，SPI 中断被触发，中断程序被执行。

因此在主机模式下使用中断驱动的 SPI 发送时，存在/SS 被置低的可能，中断应检查 MSTR 位是否被设置，一旦发现 MSTR 位被从机清 0，则必须被用户再设置。

当 SPI 被配置为从机时，/SS 引脚应为输入。当/SS 被置低时，SPI 被触发且 MISO 变为输出（如果被用户配置为输出的话）。在/SS 被置低后，其余引脚都为输入。所有引脚都为输入，而且 SPI 是被动的，这意味着它不会接收输入的数据。

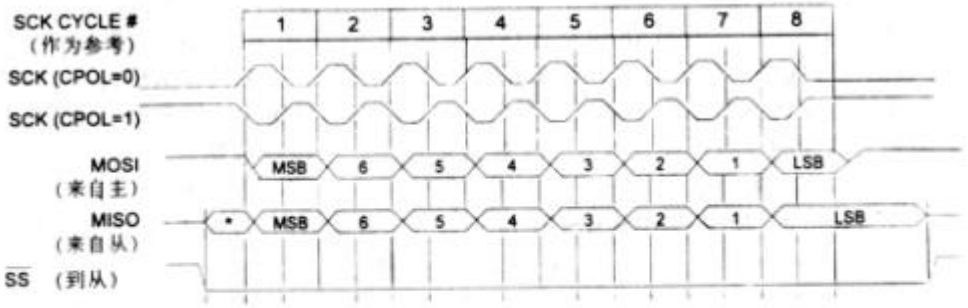
二、　数据模式

相应于串行数据，SCK 相和极性有 4 种组合，由控制位 CPHA 和 CPOL 来决定。 SPI 的传达格式如图 2. 29 和 2. 30 所示。



* 不定义，但在接收时字符的最高有效位正常。

图 2.29　当 CPHA = 0 时，SPI 传送格式



* 不定义，但在发送时，先前的最低有效位正常。

图 2.30　当 CPHA = 1 时，SPI 传送格式

三、SPI 控制寄存器——SPCR

位	7	6	5	4	3	2	1	0	
$0D（$2D）	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值:	$00								

位 7——SPIE：SPI 中断触发

如果全局中断触发，该位导致设置 SPSR 寄存器的 SPIF 位来执行 SPI 中断。

位 6——SPE：SPI 触发

当该位设置时，SPI 触发，要触发 SPI 任何操作必须设置该位。

位 5——DORD：数据的顺序

当 DORD 位被置 1 时，数据的 LSB（低位）被首先传送。

当 DORD 位被置 0 时，数据的 MSB（高位）被首先传送。

位 4——MSTR：主机/从机选择

当设置为 1 时选择主机 SPI 模式，当设置为 0 时选择从机 SPI 模式。如果/SS 被设置为输入且在 MSTR 被设置时被置低，则 MSTR 将被清除。而当 SPSR 中的 SPIF 位被设置，用户应该设置 MSTR，再触发 SPI 主机模式。

位 3——CPOL：时钟极性

当该位被置 1 时，SCK 在闲置时是高电平；当 CPOL 被清 0 时，SCK 在闲置时是低电平，详见图 2．29 和图 2．30。

位 2——CPHA：时钟相位

该位的功能详见图 2．29 和图 2．30。

位 1，0——SPR1，SPR0：SPI 时钟速率选择 1 和 0

这两位控制主机模式下器件 SCK 的速率，SPR1 和 SPR2 对于从机无影响，SCK 和振荡器频率 f_{CL} 之间的关系如表 2.15 所示。

表 2．15　SCK 和振荡器频率之间关系

SPRI	SPR0	SCK 频率	SPRI	SPR0	SCK 频率
0	0	$f_{CL}/4$	1	0	$f_{CL}/64$
0	1	$f_{CL}/16$	1	1	$f_{CL}/128$

四、SPI 的状态寄存器——SPSR

位	7	6	5	4	3	2	1	0
$0E（$2E）	SPIF	WCOL	--	--	--	--	--	SPSR
读／写:	R/W	R/W	R	R	R	R	R	R
初始化值:	$00							

位 7——SPIF：SPI 中断标志位

当单行传送完成时，SPIF 位被设置 1，且若 SPCR 中的 SPIE 被设置 1 和全局中断触发，则生成中断。如果 SS 被设置为输入且在 SPI 是主机模式时被置低，这将设置 SPIF 标志。SPIF 位在执行相应中断向量时被硬件清除。可选的，在首次读 SPI 状态寄存器时，如果 SPIF 为 1，则先清除它再访问 SPI 数据寄存器（SPDR）。

位 6——WCOL：写冲突位

如果在数据传送中 SPI 数据寄存器（SPDR）被写入。则设置 WCOL 位。在数据传送中，SPDR 寄存器读出的结果是不正确的，写入也没有反应。在首次读 SPI 状态寄存器时，如果 WCOL 为 1，则先清除它再访问 SPI 数据寄存器。

位 5~0——保留位

在 90S8515 中这几位保留，读出为 0。

90 系列单片机的 SPI 接口也被用于程序存储器和数据 E2PROM 的编程下载和上载，详见串行编程和校验部分。

五、SPI 数据寄存器——SPDR

位	7	6	5	4	3	2	1	0
$0E（$2E）	MSB						LSB	SPDR
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始化值:	$00							

SPI 数据寄存器可以读／写，用于在寄存器堆和 SPI 移位寄存器之间传送数据。写入该寄存器时初始化数据传送，读该寄存器时读到的是移位寄存器接收缓冲区的值。

2.9.2　通用串行接口 UART

90 系列单片机带有一个全双工的通用串行异步收发器（UART），主要特征如下：（1）波特率发生器可以生成任何波特率。

（2）在 XTAL 低频率下有高的波特率。
（3）8 位和 9 位数据。
（4）噪声滤波。
（5）超越误差的探测。
（6）帧错误探测。
（7）错误起始位的探测。
（8）三个独立的中断，TX 完成，TX 数据寄存器为空，RX 完成。

一、数据传送

UART 传送器的方框示意图见图 2．31。数据传送通过把被传送的数据写入 UART I／O 寄存器 UDR 来初始化，在以下情况数据从 UDR 传送到移位寄存器中。

（l）当前一个字符的停止位被移出后新的字符被写入 UDR 寄存器，移位寄存器立即再被装入；

（2）当前一个字符的停止位被移出前新的字符被写入 UDR 寄存器，移位寄存器在当前字符的停止位移出后被装入。

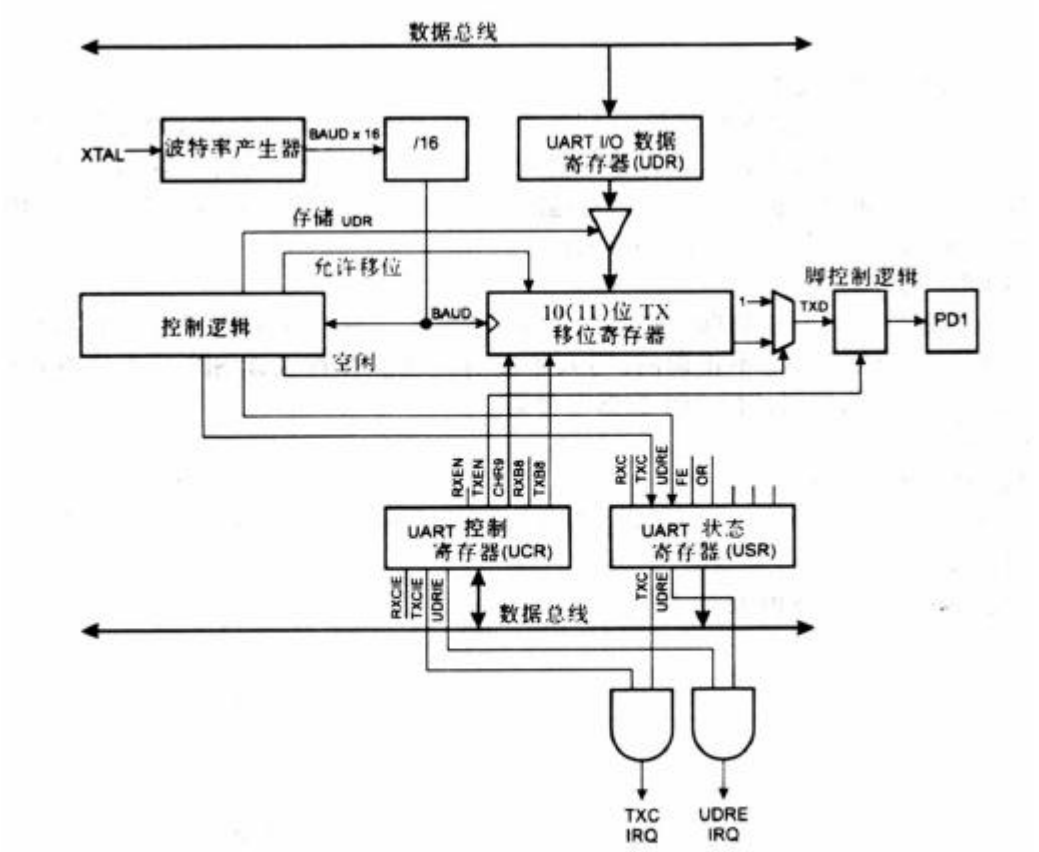


图 2.31　UART 传送器方框示意图

如果 10（11）位传送移位寄存器是空的，或当数据从 UDR 中传送到移位寄存器时，UART 状态寄存器，USR 的 UDRE 位（UART 状态寄存器空）被设置。当这位被设置为 1 时，UART 准备接收下一个字符；当数据从 UDR 传送到 10 位（11 位）的移位寄存器中时移位寄存器的第 0 位（起始位）清 0 而第 9 或第 10 位置 1（停止位）。如果选择 9 位数据（UART 控制寄存器——UCR 的 CHR9 位置位），UCR 的 TXB8 位被传送到移位寄存器的第 9 位。

在波特率时钟加载到移位寄存器的传送操作时，起始位从 TXD 引脚移出，然后是数据，最低位在先。当停止位被移出时，如果在传送中有新数据被写入 UDR 中，则被装入移位寄存器中。在装入过程中，UDRE 被设置。如果在停止位被移出时 UDR 寄存器中没有新的数据，UDRE 标志位将保持为 1 直到 UDR 被重写。当没有新的数据被写入时，而且停止位在 TXD 上保持了一位的长度，USR 的 TX 完成的标志位——TXC 被置位。

当 UCR 中的 TXEN 位被设置为 1 时允许 UART 传送，通过清除该位，PD1 引脚可以被用于通用的 I/O。当 TXEN 被设置时，UART 将被连到 PD1 引脚，而不管 DDRB 中的 DDD1 位的是否设置。

二、数据接收

图 2.32 为 UART 的接收器的示意图。

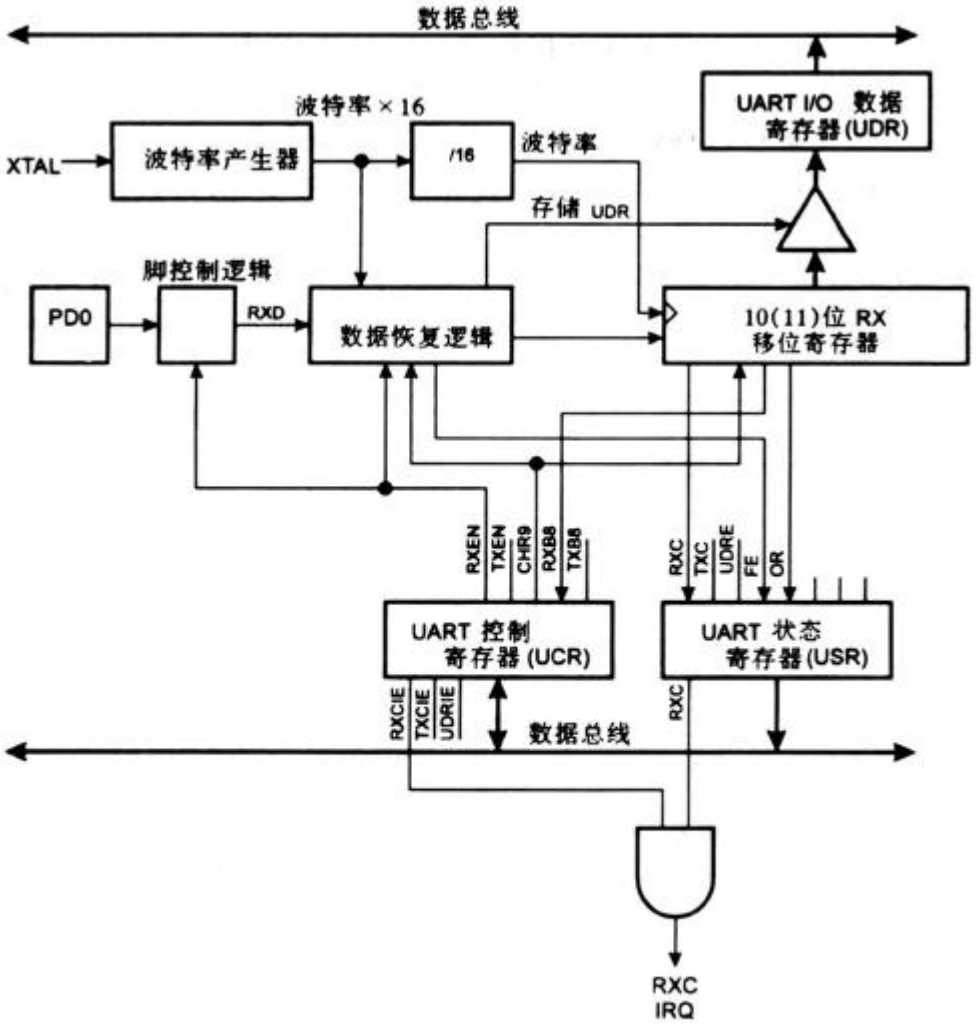


图 2.32　UART 的接收器

接收器前端的逻辑以 16 倍波特率采样 RXD 引脚的信号。当线路闲置时，一个逻辑 0 的采样将被转换为起始位的下降沿，并且起站位的探测序列被初始化。让采样 1 指示出第一个 0 采样，跟随 1 到 0 的转换之后，接收器在第 8、9 和 10 个采样点采样 RXD 引脚。如果三个采样中两个或两个以上是逻辑 1，则起始位是噪声尖峰而被拒绝，接收器继续探测下一个 1 到 0 的转换。

如果一个有效的起始位被发现，就开始起始位之后的数据位的采样，这些位也在第 8、9 和 10 处采样，3 取 2 作为该位的逻辑值，在采样的同时这些位被移入传送移位寄存器。采样输入的

字符如图 2—33 所示。

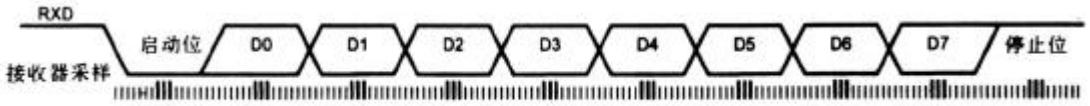


图 2.33　接收器数据采样

当停止位到来时，三个采样中的大数应为 1 才能接受该停止位。如果两个或更多为逻辑 0，UART 状态寄存器（USR）的帧错误（FE）标志被设置 1，在读 UDR 寄存器之前，用户应检查 FE 帧错误标志。一旦或无效的停止位在字符接收周期的结束时被收到，数据被传送到 UDR 寄存器而 USR 的 RXC 标志位被设置。UDR 实际上是两个物理分离的寄存器，一个发送数据，一个接收数据。当读 UDR 时，接收数据寄存器被访问；当写 UDR 寄存器时，传送数据寄存器被访问。如果选择了 9 位数据（UART 控制寄存器 UCR 中的 CHR9 位被设置），在数据被传送到 UDR 时，传送移位寄存器的第 9 位被装入到 UCR 的 RXB8 位。

如果在收到一个字符的最后一个接收后 UDR 寄存器还没有被读走，UCR 中的超越误差标志（OR）被设置，这意味着移入移位寄存器的最后的数据字节不能被送到 UDR 中而丢失。OR 位被缓冲，当 UDR 中的有效的数据字节被读出时该位被更新。因此，用户在读 UDR 后应检查 OR 位来探测任何的超越错误。

通过清除 UCR 寄存器中的 RXEN 位使接收器被禁止，这意味着 PD0 可以被用做通用的 I／O 引脚，当 RXEN 被设置， USRT 接收器被连到 PD0 引脚而不管 DDRB 中的 DDD0 位是否设置。

三、 UART 控制

1. UART I／O 数据寄存器——UDR

位	7	6	5	4	3	2	1	0
$0C（$2C）	MSB						LSB	UDR
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始化值:	$00							

UDR 寄存器是两个物理分离的寄存器分享相同的 I／O 地址。当写人寄存器时，UART 的发送数据寄存器被写入；当读 UDR 时，读的是 UART 接收寄存器。

2. UART 状态寄存器——USR

位	7	6	5	4	3	2	1	0
$0B（$2B）	RXC	TXC	UDR	FE	OR	--	--	USR
读／写:	R/W	R/W	R/W	R/W	R/W	R	R	R
初始化值:	$00							

USR 寄存器是一个只读的寄存器，提供 UART 的状态信息。

位 7——RXC：UART 接收完成

当收到的字符从接收移位寄存器传到 UDR 中时该位被设置。不论探测到任何的帧错误，该位都被设置。当 UCR 中的 RXCIE 位被设置后，UART 接收完成中断将被执行（当 RXC 被设置），RXC 在读 UDR 时被清除。当使用中断数据接收时，接受完成中断子程序必须读 UDR 而清除 RXC，否则在于程序完成时会引起新的中断。

位 6——TXC：UART 接收完成

当发送移位寄存器的全部数据被移出后且没有新的数据被写入 UDR 时该位被设置。这个标志位在半双工的通讯接口中很有用。当完成发送后立即释放通讯总线，并必须进入接收模式。当 UCR 中的 TXCIE 位被设置后，设置 TXC 将导致 UART 发送完成中断被执行，TXC 在执行相应的中断向量时被硬件清除。可选择的，TXC 位也可以通过在该位写一个逻辑 1 而被清除。

位 5——UDRE：UART 数据寄存器空

当写入 UDR 的字符被传送到发送移位寄存器中时该位被设置，设置该位指示出发送器准备新的数据发送。当 UCR 中的 UDRIE 位被设置时，UART 发送完成中断将被执行。只要 UDRE 被设置，UDRE 可以通过写 UDR 而清除。当使用中断驱动的数据发送时，UART 数据寄存器空的中断服务程序应该写 UDR 来清除 UDRE，否则在中断子程序完成时将发生新的中断。在复位时， UDRE 被设置为 1 指示出准备传送。

位 4——FE：帧出错

如果在帧出错条件被检测到时该位被设置（如当收到数据的停止位为 0 时）。FE 在收到数据的停止位为 1 时被清除。

位 3—OR：超越出错

如果超越出错条件被检测到，该位被设置（如当 UDR 寄存器中的数据没有在新的数据被移入接收移位寄存器之前被读走）。OR 位被缓冲，这意味着一旦 UDRE 中有效的数据被读走后它将被设置。OR 位在收到的数据被传送到 UDR 中时被清除。

位 2~0——Res:保留位

在 90S8515 中这些位被保留，读出为 0。

3. UART 控制寄存器——UCR

位	7	6	5	4	3	2	1	0
$0A（$2A）	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
读／写:	R/W	R/W	R/W	R/W	R/W	R/W	R	W
初始化值:	$00							

位 7——RXCIE：RX 完成中断触发

当该位被置 1 时能，如果全局中断被触发，在 USR 中设置 RXC 位将导致接收完成中断被执行。

位 6——TXCIE：TX 完成中断触发

当该位被置 1 时，如果全局中断被触发，在 USR 中设置 TXC 位将导致发送完成中断被执行。

位 5——UDRIE：UART 数据寄存器空中断触发

当该位被置 1 时，如果全局中断被触发，在 USR 中设置 UDRE 位将导致 UART 数据寄存器空中断被执行。

位 4—RXEN：接收触发

当该位被设置时允许 UART 接收。当接收器被禁止时，TXC、OR 和 FE 位状态标志位不能设置。如果这些位被设置，在把 RXEN 关闭时不能清除它们。

位 3——TXEN：发送触发

当该位被设置为 1 时允许 UART 发送。如在发送数据时禁止发送器，则在移位寄存器的数据和后续 UDR 中的数据被全部发送完成之前发送器不会被禁止。

位 2—CHR9：9 位字符

当设置该位时，发送和接收的数据是 9 位加上起始和停止位。第 9 位通过 UCR 中的 RXB8 和 TXB8 位来分别读和写。第 9 位可以作为额外的停止位和奇偶位。

位 1——RXB8：收到的数据第 8 位

当 CHR9 被设置时，RXB8 是收到数据的第 9 数据位。

位 0——TXB8：发送的数据第 8 位

当 CHR9 被设置时，TXB8 是发送数据的第 9 数据位。

4。波特率发生器

波特率发生器是依据以下等式的分频器来产生波特率：

$$BAUD = FCK / [16(UBRR + 1)]$$

其中 BAUD 表示波特率；FCK 表示晶振频率；UBRR 表示 UART 波特率寄存器的值，UBRR(0~255)。

对于标准的晶振频率，可以通过表 2.16 来设置 UBRR 而产生常用的波特率，生成与实际波特率相差小于 2%的 UBRR 的值。

表 2.16　不同晶振频率的 UBRR 设置

波特率	1 MHz	误差%	1.843 MHz	误差%	2 MHz	误差%	2.457 6 MHz	误差%
2400	UBRR= 25	0.2	UBRR= 47	0.0	UBRR= 51	0.2	UBRR= 63	0.0
4 800	UBRR= 12	0.2	UBRR= 23	0.0	UBRR= 25	0.2	UBRR= 31	0.0
9 600	UBRR= 65	7.5	UBRR= 11	0.0	UBRR= 12	0.2	UBRR= 15	0.0
14 400	UBRR= 3	7.8	UBRR= 7	0.0	UBRR= 8	3.7	UBRR= 10	3.1
19 200	UBRR= 2	7.8	UBRR= 5	0.0	UBRR= 6	7.5	UBRR= 7	0.0
28 800	UBRR= 1	7.8	UBRR= 3	0.0	UBRR= 3	7.8	UBRR= 4	6.3
57 600	UBRR= 0	7.8	UBRR= 1	0.0	UBRR= 1	7.8	UBRR= 2	12.5
115 200	UBRR= 0	84.3	UBRR= 0	0.0	UBRR= 0	7.8	UBRR= 0	25.0
波特率	3.276 8 MHz	误差%	3.686 4 MHz	误差%	4 MHz	误差%	4.608 MHz	误差%
2 400	UBRR= 84	0.4	UBRR= 95	0.0	UBRR= 103	0.2	UBRR= 119	0.0
4 800	UBRR= 42	0.8	UBRR= 47	0.0	UBRR= 51	0.2	UBRR= 59	0.0
9 600	UBRR= 20	1.6	UBRR= 23	0.0	UBRR= 25	0.2	UBRR= 29	0.0
14 400	UBRR= 13	1.6	UBRR= 15	0.0	UBRR= 16	2.1	UBRR= 19	0.0
19 200	UBRR= 10	3.1	UBRR= 11	0.0	UBRR= 12	0.2	UBRR= 14	0.0
28 800	UBRR= 6	1.6	UBRR= 7	0.0	UBRR= 8	3.7	UBRR= 9	0.0
57 600	UBRR= 3	12.5	UBRR= 3	0.0	UBRR= 3	7.8	UBRR= 4	0.0
115 200	UBRR= 1	12.5	UBRR= 1	0.0	UBRR= 1	7.8	UBRR= 2	20.0
波特率	7.372 8 MHz	误差%	8 MHz	误差%	9.216 MHz	误差%	11.059 MHz	误差%
2 400	UBRR= 191	0.0	UBRR= 207	0.0	UBRR= 239	0.0	UBRR= 287	0.0
4 800	UBRR= 95	0.0	UBRR= 103	0.0	UBRR= 119	0.0	UBRR= 143	0.0
9 600	UBRR= 47	0.0	UBRR= 51	0.0	UBRR= 59	0.0	UBRR= 71	0.0
14 400	UBRR= 31	0.0	UBRR= 34	0.0	UBRR= 29	0.0	UBRR= 47	0.0
19 200	UBRR= 23	0.0	UBRR= 25	0.0	UBRR= 19	0.0	UBRR= 35	0.0
28 800	UBRR= 15	0.0	UBRR= 16	0.0	UBRR= 14	0.0	UBRR= 23	0.0
57 600	UBRR= 7	0.0	UBRR= 8	0.0	UBRR= 9	0.0	UBRR= 11	0.0
115 200	UBRR= 3	0.0	UBRR= 3	0.0	UBRR= 4	0.0	UBRR= 5	0.0
波特率	14.746 MHz	误差%	16 MHz	误差%	18.432 MHz	误差%	20 MHz	误差%
2 400	UBRR= 383	–	UBRR= 416	–	UBRR= 479	–	UBRR= 520	–
4 800	UBRR= 191	0.0	UBRR= 207	0.2	UBRR= 239	0.0	UBRR= 257	–
9 600	UBRR= 95	0.0	UBRR= 103	0.2	UBRR= 119	0.0	UBRR= 129	0.2
14 400	UBRR= 63	0.0	UBRR= 68	0.6	UBRR= 79	0.0	UBRR= 86	0.2
19 200	UBRR= 47	0.0	UBRR= 51	0.2	UBRR= 59	0.0	UBRR= 64	0.2
28 800	UBRR= 31	0.0	UBRR= 34	0.8	UBRR= 39	0.0	UBRR= 42	0.9
57 600	UBRR= 15	0.0	UBRR= 16	2.1	UBRR= 19	0.0	UBRR= 21	1.4

5. 波特率寄存器——UBRR

位	7	6	5	4	3	2	1	0	
$09($29)	MSB							LSB	UBRR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值	$00								

UBRR 是 8 位可以读/写的寄存器，用来确定波特率。

2.10　AVR 单片机模拟比较器

2.10.1 模拟比较器

模拟比较器对正极 PB2 引脚（AIN0）和负极 PB3 引脚（AIN1）之上的输入值进行比较。当 PB2 上的电压高于 PB3 的电压时，模拟比较器输出 ACO 被设为 1。比较器的输出可以被设置为触发定时／计数器 1 的输入捕获功能，此外，比较器的输出可以被设置为触发一个独立于模拟比较器的中断。用户可以选择比较器输出上升、下降，或触发的中断触发。比较器的方框图和周围电路如图 2．34 所示。

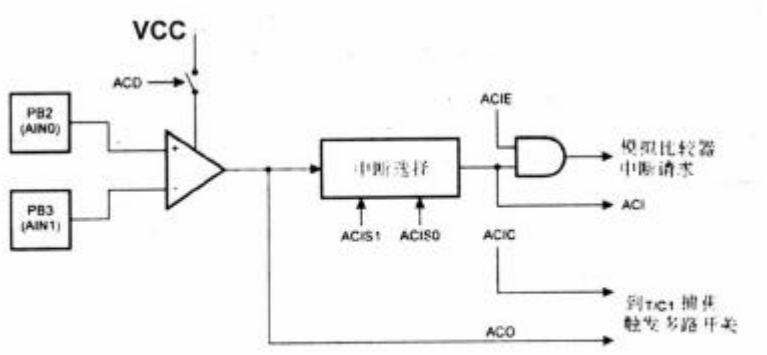


图 2.34　模拟比较器方框图

2.10.2 模拟比较器控制和状态寄存器 ACSR

位	7	6	5	4	3	2	1	0	
$08($28)	ACD	-	ACO	ACI	ACIE	ACIC	ACS1	ACIS0	ACSR
读／写	R／W	R	R	R／W	R／W	R／W	R/W	R/W	
初始化值	$00								

位 7——ACD：模拟比较器禁止

当该位设为 1 时，模拟比较器的电源关闭。该位可以在任何时候被设置，以便关闭模拟比较器。在闲置模式的电源消耗为临界时，它最为常用，且无需从模拟比较器唤醒。改变 ACD 位时，模拟比较器中断必须通过清空 ACSR 中的 ACIE 位来禁止；否则，在该位改变时，会产生中断。

位 6——Res：保留位

90 系列单片机的该位为保留位，总读 0。

位 5——ACO：模拟比较器输出

ACO 直接与模拟比较器的输出相连。

位 4——ACI：模拟比较器中断标志位

当比较器输出事件触发由 ACIl 和 ACIO 定义的中断模式时，这一位被设为 1。若 ACIE 位被设为 1，且 SREG 中的 I 位被设为 1 时，执行模拟比较器的中断程序。当执行相应的中断处理向量时，ACI 被硬件清空。作为替换，ACI 通过对标志位写入逻辑 1 来清空。

位 3——ACIE：模拟比较器中断触发

当 ACIE 位设为 1，且状态寄存器中的 I 位被设为 1 时，模拟比较器中断被触发。当被清为 0

时，中断被禁止。

位 2——ACIC：模拟比较器输入捕获触发

当设置为 1 时，该位触发定时计数器 1 的输入捕获功能，由模拟比较器来触发。在这种情况下，模拟比较器的输出直接连到输入捕获前端逻辑，使比较器能利用定时器 / 计数器 1 输入捕获中断的噪声消除和边缘选择的特性。当该位被清除时，模拟比较器和输入捕获功能之间没有联系。为了使比较器触发定时器 / 计数器 1 的输入捕获中断，定时器中断屏蔽寄存器（TIMSK）的 TICIE1 位必须被设置。

位 1，0——ACIS1,ACIS0：模拟比较器中断模式选择

这些位决定了哪一比较器事件触发模拟比较器中断。表 2．17 所示为不同的设置。

表 2.17　ACIS1/ACIS0 设置

ACIS1	ACIS0	中断模式	ACIS1	ACIS0	中断模式
0	0	输出触发，比较器中断	1	0	下降输出沿，比较器中断
0	1	与上相反	1	1	上升输出沿，比较器中断

注意：当改变 ACIS1 / ACISO 位时，必须通过清除 ACSR 寄存器中的中断触发位来禁止模拟比较器中断。否则，当这些位改变时会发生中断。

2．11　AVR 单片机 I/O 端口

2.11.1　端口 A

一、A 口特性

A 口为一个 8 位的双向 I / O 口。

A 口分配有 3 个数据存储地址，分别为数据寄存器 PORTA $1B（$3B）、数据方向寄存器 DDRA $1A（$3A）和 A 口的输出引脚 PINA $19（$39）。A 口的输入引脚地址为只读，而数据寄存器和数据方向寄存器为可读写。

所有的 A 口引脚都有独立可选的上拉，A 口输出缓冲器可以吸收 20 mA 的电流以直接驱动 LED 显示。当 PA0 到 PA7 引脚被用作输入且被外部拉低时，若内部拉高被触发，这些引脚将成为电流源（I_{IL}）。

A 口引脚具有与可选的外部数据 SRAM 有关的第二功能，A 口在访问外部数据存储器时可以配置为复用的低位地址 / 数据线，在该模式下，A 口有内部的上拉。

当通过 MCU 控制寄存器 MCUCR 的 SRE，外部 SRAM 触发位把 A 口设置为第二功能，更改的设置会覆盖数据方向存储器。

1.　A 口数据寄存器——PORTA

位	7	6	5	4	3	2	1	0	
$1B($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
读 / 写	R / W	R/W	R / W	R / W	R / W	R / W	R/W	R / W	
初始化值	$00								

2. A 口数据方向寄存器——DDRA

位	7	6	5	4	3	2	1	0	
$1A($3A)	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值	$00								

3. A 口输入脚地址——PINA

位	7	6	5	4	3	2	1	0	
$1A($3A)	PINA7	PINA 6	PINA 5	PINA 4	PINA 3	PINA 2	PINA 1	PINA 0	PINA
读/写	R	R	R	R	R	R	R	R	
初始化值	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

A 口的输入引脚地址 PINA 不是一个寄存器，该地址允许对 A 口的每一个引脚的物理值进行访问。当读 PORTA 时，读到的是 PORTA 的数据锁存器；当读 PINA 时，引脚上的逻辑值被读取。

二、A 口作为通用的数字 I/O

当作为数字 I/O 口时 A 口所有的 8 位都等效。

PAn 为通用 I/O 引脚：DDRA 寄存器的 DDAn 位选择引脚的方向。如果 DDAn 设为 1，PAn 被配置为输出引脚；如果 DDAn 设为 0，PAn 被配置为输入引脚；如果 PORTAn 被设置为 1，DDAn 被配置为输入引脚，则 MOS 上拉电阻被触发。为了关断上拉电阻，PORTAn 位必须被清除或者引脚被配置为输出引脚。 A 口引脚 DDAn 的作用见表 2．18。

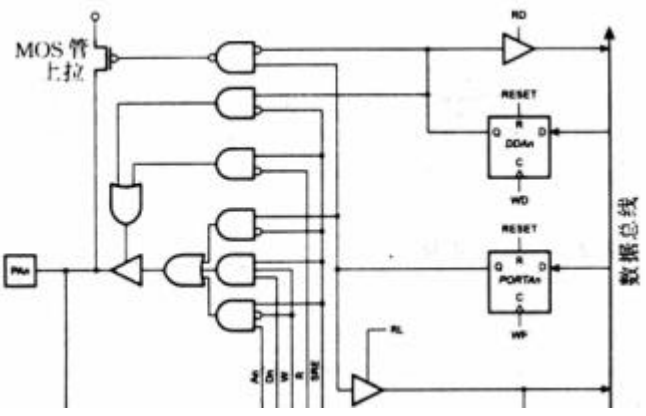
表 2.18　A 口引脚的 DDAn 的作用

DDAn	PORTAn	I/O	上拉	注　　释
0	0	输入	否	三态(高阻)
0	1	输入	是	上拉低 PAn 脚输出电流
1	0	输出	否	推挽 0 输出
1	1	输出	否	推挽 1 输出

n:7,6,5,…,0 为引脚数。

三、A 口原理图

A 口原理如图 2．35 所示（注意：所有引脚是同步的，同步锁存器在图中并未列出）。



2.11.2　端口 B

一、B 口特性

B 口为一个 8 位的双向 I／O 口。

B 口分配有 3 个数据存储地址，分别为数据寄存器 PORTB $18（$38）、数据方向寄存器 DDRB $17（$37）和 B 口的输出引脚 PINB $16（$36）。B 口的输入引脚地址为只读，而数据寄存器和数据方向寄存器为可读写。

所有的 B 口引脚均有单独的可选择拉高。 B 口输出缓冲器可以吸收 20 mA 的电流以直接驱动 LED 显示。当 PB0 到 PB7 引脚被用作输入。且被外部拉低时，若内部拉高被触发，这些引脚将成为电流源（I_{IL}）。

具有第二功能的 B 口引脚如表 2.19 所示。当引脚被用作第二功能时，DDRB 和 PORTB 寄存器必须根据第二功能说明来设置。

表 2.19　B 口引脚第二功能

口引脚	第　二　功　能	口引脚	第　二　功　能
PB0	T0(定时器/计数器 0 外部计数器输入)	PB4	$\overline{SS}$(SPI 从选择输入)
PB1	T1(定时器/计数器 1 外部计数器输入)	PB5	MOSI(SPI 总线主输出/从输入)
PB2	AIN0(模拟比较器正输入)	PB6	MOSO(SPI 总线主输出/从输入)
PB3	AIN1(模拟比较器负输入)	PB7	SCK(SPI 总线串行时钟)

1.　B 口数据寄存器——PORTB

位	7	6	5	4	3	2	1	0	
$18($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
读／写	R／W	R/W	R／W	R／W	R／W	R／W	R/W	R／W	
初始化值	$00								

2.　B 口数据方向寄存器——DDRB

位	7	6	5	4	3	2	1	0	
$17($37)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
读／写	R／W	R/W	R／W	R／W	R／W	R／W	R/W	R／W	
初始化值	$00								

3.　B 口输入引脚地址——PINB

位	7	6	5	4	3	2	1	0	
$16($36)	PINB7	PINB 6	PINB 5	PINB 4	PINB 3	PINB 2	PINB 1	PINB 0	PINB
读／写	R	R	R	R	R	R	R	R	
初始化值	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

B 口输入引脚地址 PINB 并不是一个寄存器，这一地址允许对 B 口每个引脚的物理值进行访问。当读取 PORTB 时，PORTB 数据锁存器被读取，当读取 PINB 时，引脚上的当前逻辑值被读取。

二、 B 口作为通用数字 I／O

当 B 口上的所有 8 个位用作数字 I / O 引脚时功能一样。

PBn 为通用 I / O 引脚：DDRB 寄存器中的 DDBn 位选择该引脚的方向。若 DDBn 被设为 1 时，PBn 设为输出引脚。若 DDBn 被清为 0 时，PBn 设为输入引脚。若 PORTBn 被设为 1，引脚被设为输入时，MOS 拉高电阻被触发。为关闭拉高电阻，PORTBn 必须被清为 0，或者该引脚必须被设置为输出引脚。 B 口引脚的 DDBn 作用见表 2. 20。

表 2.20　B 口引脚的 DDBn 作用

DDBn	PORTBn	I/O	上拉	注释
0	0	输入	否	三态(高阻)
0	1	输入	是	上拉低 PBn 脚输出电流
1	0	输出	否	推挽 0 输出
1	1	输出	否	推挽 1 输出

n：7,6,…,0 为引脚数。

三、 B 口的可选择性功能：

SCK——B 口，位 7

SCK、SPI 的主时钟输出、从时钟输入。当 SPI 被触发为从机时，该引脚被作为输入而不管 DDB7 的设置；当 SPI 被触发为主机时，该引脚的数据方向由 DDB7 来控制；当该引脚被强制作为输入时，内部的上拉仍可以被 PORTB7 位来控制，详见 SPI 口的描述。

MISO——B 口，位 6

MISO、SPI 的主数据输入、从时的数据输出。当 SPI 被触发为主机时，该引脚被作为输入而不管 DDB6 的设置；当 SPI 被触发为从机时，该引脚的数据方向由 DDB6 来控制；当该引脚被强制作为输入时，内部的上拉仍可以被 PORTB6 位来控制，详见 SPI 口的描述。

MOSI-B 口，位 5:

MOSI、SPI 的主数据输出、从时的数据输入。当 SPI 被触发为从机时，该引脚被作为输入而不管 DDB5 的设置；当 SPI 被触发为主机时，该引脚的数据方向由 DDB5 来控制；当该引脚被强制作为输入时，内部的上拉仍可以被 PORTB5 位来控制，详见 SPI 口的描述。

/SS—B 口,位 4:

/SS 为从机端口选择信号输入端。当 SPI 被触发为从机时,该引脚被作为输入而不管 DDB5 的设置。作为从机时，当该引脚被置低时 SPI 被触发。当 SPI 被触发为主机时，该引脚的数据方向由 DDB5 来控制。当该引脚被强制作为输入时,内部的上拉仍可以被 PORTB4 位来控制，详见 SPI 口的描述。

AIN1—B 口,位 3:

AIN1 为模拟比较器负极输入。当被设置为输入（DDB3 被清为 0），且内部 MOS 拉高电阻关闭（PB3 被清为 0）时，该引脚用作片内模拟比较器的负极输入。

AIN0—B 口,位 2:

AIN0 为模拟比较器正极输入。当被设置为输入（DDB2 被清为 0），且内部 MOS 拉高电阻关闭（PB2 被清为 0）时，该引脚用作片内模拟比较器的正极输入。

T1—B 口,位 1:

T1 为定时器 / 计数器 1 的计数输入源，详见定时器部分。

T0—B 口,位 0:

T0 为定时器 / 计数器 0 的计数输入源，详见定时器部分。

四、 B 口原理国

B 口原理图如图 2.26~图 2.41 所示，注意：所有引脚为同步的。同步锁存器图中并未列出。

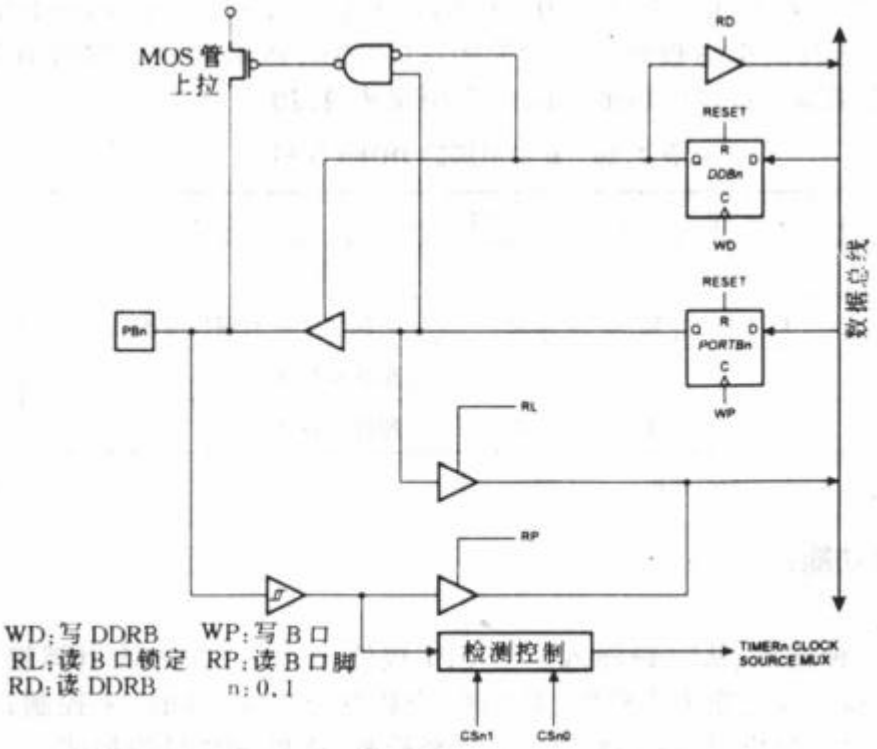


图 2.36　B 口原理图(PB0 和 PB1 脚)

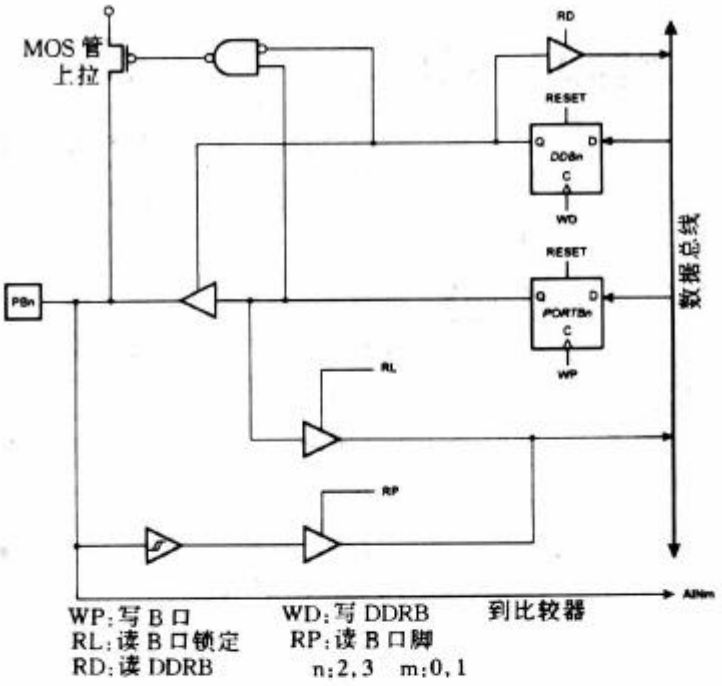


图 2.37　B 口原理图(PB2 和 PB3 脚)

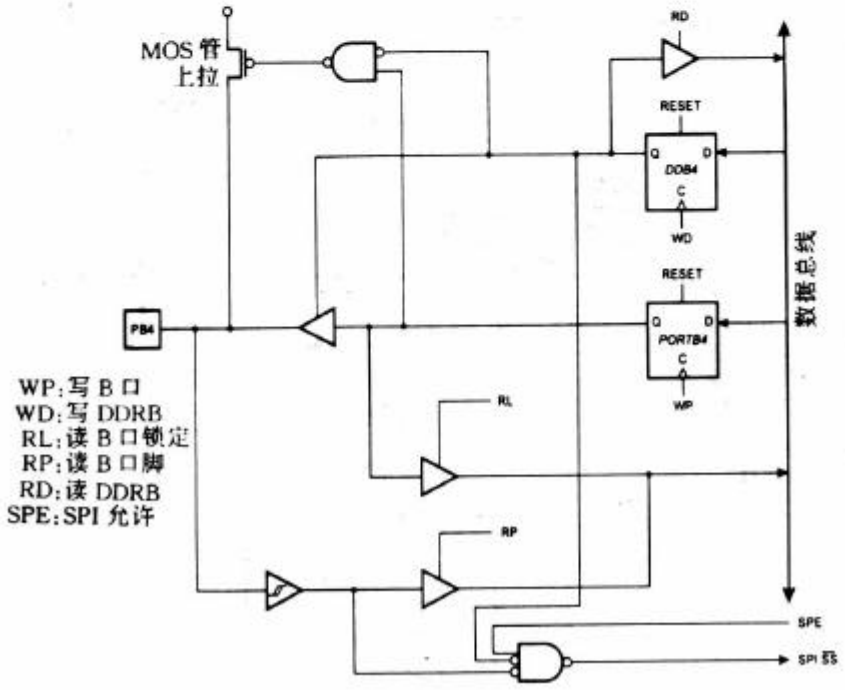


图 2.38　B 口原理图(PB4 脚)

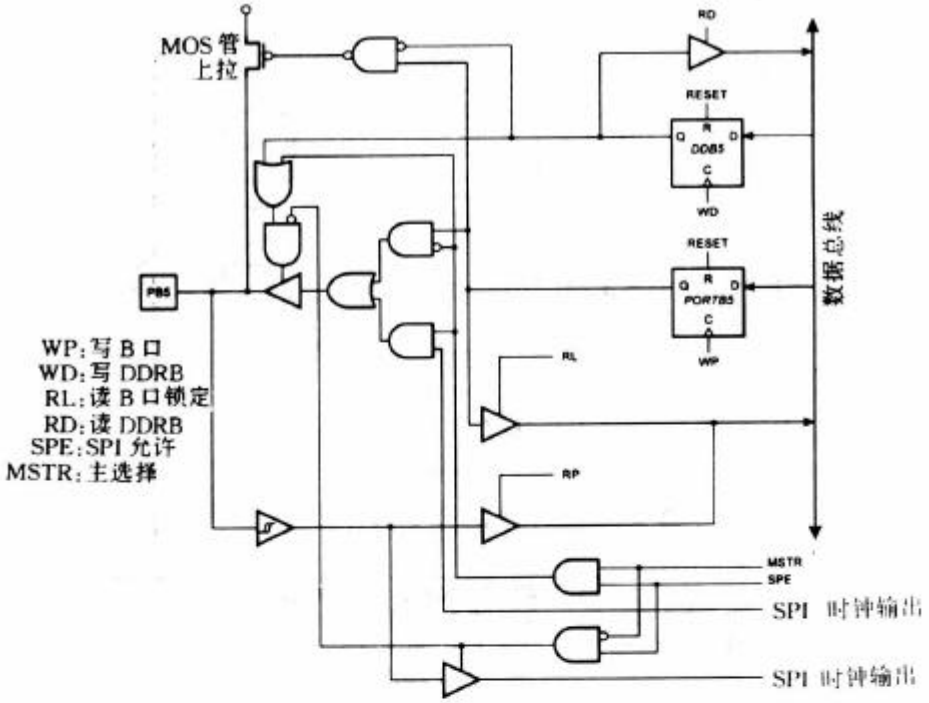


图 2.39　B 口原理图(PB5 脚)

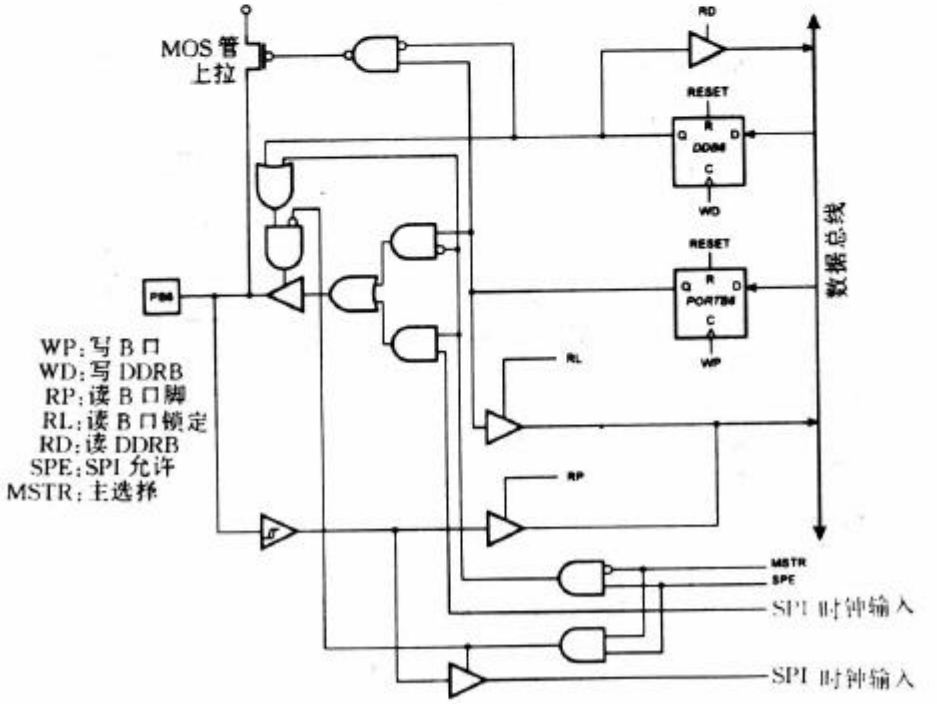


图 2.40　B 口原理图(PB6 脚)

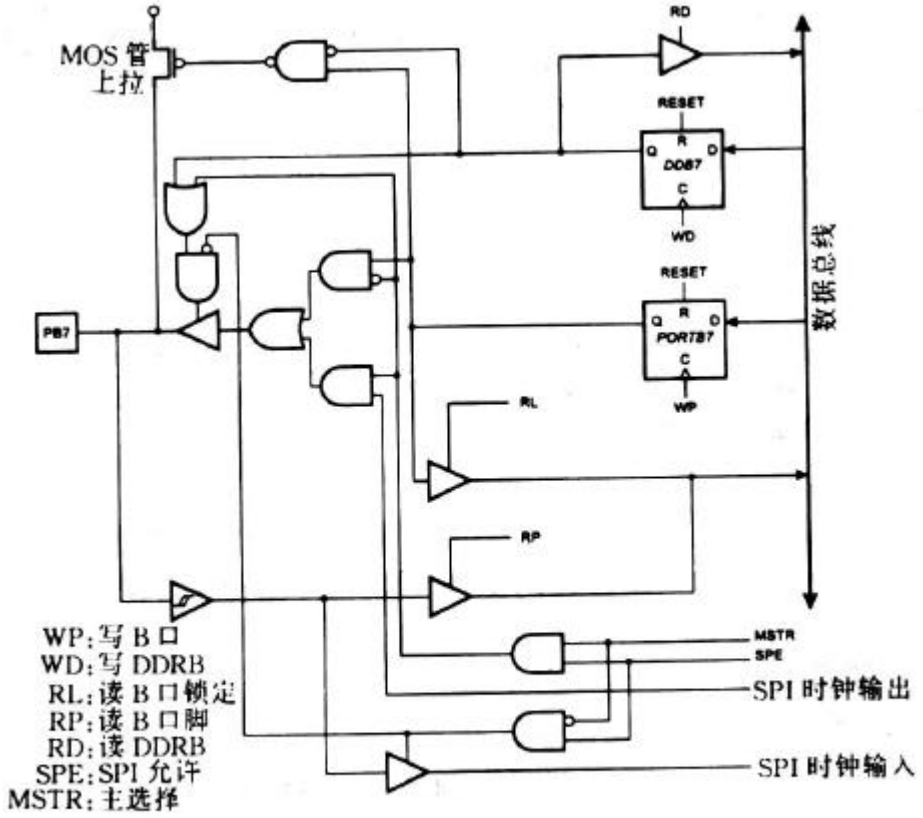


图 2.41　B 口原理图(PB7 脚)

2.11.3 端口 C

一、C 口

C 口为一个 8 位的双向 I／O 口。

C 口分配有 3 个数据存储地址，分别为数据寄存器 PORTC $15（$35）、数据方向寄存器 DDRC $14（$34）和 C 口的输出引脚 PINC $13（$33）。C 口的输入引脚地址为只读，而数据寄存器和数据方向寄存器为可读写。

所有的 C 口引脚均有单独的可选择拉高。 C 口输出缓冲器可以吸收 20MA 的电流以直接驱动 LED 显示。当 PC0 到 PC7 引脚被用作输入且被外部拉低时，若内部拉高被触发，这些引脚将成为电流源（I_{IL}）。

C 口引脚具有与可选的外部数据 SRAM 有关的第二功能， C 口在访问外部数据存储器时可以配置为复用的高位地址线，在该模式下，C 回输出 1 时使用内部的上拉。

当通过 MCU 控制寄存器 MCUCR 的 SRE，外部 SRAM 触发位把 C 口设置为第二功能，更改的设置会覆盖数据方向存储器。

1 C 口数据寄存器——PORTC

位	7	6	5	4	3	2	1	0	
$15($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
读／写	R／W	R/W	R／W	R／W	R／W	R／W	R/W	R／W	
初始化值	$00								

2. C 口数据方向寄存器——DDRC

位	7	6	5	4	3	2	1	0	
$14($34)	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值	$00								

3. C 口输入引脚地址——PINC

位	7	6	5	4	3	2	1	0	
$13($33)	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
读/写	R	R	R	R	R	R	R	R	
初始化值	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

C 口的输入引脚地址 PINC 不是一个寄存器，该地址允许对端口 C 的每一个引脚进行存取。当读 PORTC 时，读到的是 PORTC 的数据锁存器；当读 PINC 时，引脚上的逻辑值被读取。

二、C 口作为通用的数字 I/O

当作为数字 I/O 口时 C 口所有的 8 位都等效。

PCn 为通用 I/O 引脚：DDRC 寄存器的 DDCn 位选择引脚的方向。如果 DDCn 设为 1，PCn 被配置为输出引脚；如果 DDCn 设为 0，PCn 被配置为输入引脚；如果 PORTCn 被设置为 1，DDCn 被配置为输入引脚，则 MOS 上拉电阻被触发，为了关断上拉电阻，PORTCn 位必须被清除或者引脚被配置为输出引脚。C 口引脚 DDCn 的作用见表 2. 21。

表 2.21　C 口引脚的 DDCn 作用

DDCn	PORTCn	I/O	上　拉	注　　释
0	0	输入	否	三态(高阻)
0	1	输入	是	上拉低 PCn 脚输出电流
1	0	输出	否	推挽 0 输出
1	1	输出	否	推挽 1 输出

n:7,…,0 为引脚数。

三、C 口原理图

C 口原理图如图 2. 42 所示。注意：所有的引脚是同步的，同步锁存器图中并未列出。

2.11.4　端口 D

一、D 口特性

D 口是一个带内部上拉的 8 位双向 I/O 口。

D 口占了 3 个数据存储器的地址，一个是数据寄存器 PORTD $12（$32）、数据方向寄存器 DDRD $11（$31）和端口 D 输入引脚 PIND $10（$30）。D 口的引脚地址是只读的，而数据寄存器和数据方向寄存器可以读写。

D 口的输出缓冲器可以吸收 20MA 的电流。D 口的引脚在触发内部上拉时，如果外部被拉低就会成为电流源(I_{IL})。

某些 D 口的引脚是有第二功能如表 2. 22 所示。

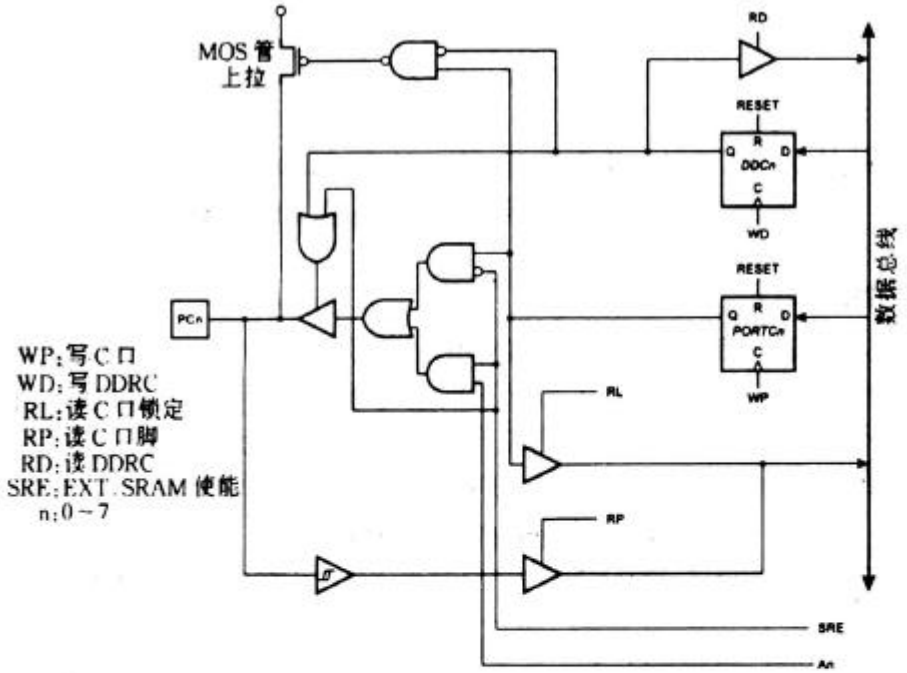


图 2.42　C 口原理图(PC0～PC7)

表 2.22　D 口引脚第二功能

口引脚	第　二　功　能	口引脚	第　二　功　能
PD0	RDX(UART 输入线)	PD5	OC1A(T/C1 输出比较 A 匹配输出)
PD1	TDX(UART 输出线)	PD6	$\overline{WR}$(写选通)
PD2	INT0(外部中断 0 输入)	PD7	$\overline{RD}$(读选通)
PD3	INT1(外部中断 1 输入		

1.　D 口数据寄存器——PORTD

位	7	6	5	4	3	2	1	0	
$12($32	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值	$00								

2.　D 口数据方向寄存器——DDRD

位	7	6	5	4	3	2	1	0	
$11($31)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始化值	$00								

3. D 口输入引脚地址——PIND

位	7	6	5	4	3	2	1	0	
$10($30)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
读/写	R	R	R	R	R	R	R	R	
初始化值	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	HI-Z	

D 口的输入引脚地址 PIND 不是一个寄存器，该地址允许对端口 D 的每一个引脚进行存取。当读 PORTD 时，读到的是 PORTD 的数据锁存器；当读 PIND 时，引脚上的逻辑值被读取。

二、 D 口作为通用的数字 I/O

PDn，通用 I/O 引脚：DDRD 寄存器的 DDDn 位选择引脚的方向。如果 DDDn 设为 1，PDn 被配置为输出引脚，如果 DDDn 设为 0，PDn 被配置为输入引脚；如果 PORTDn 被设置为 1，DDDn 被配置为输入引脚，则 MOS 上拉电阻被触发。为了关断上拉电阻，PORTDn 位必须被清除或者引脚被配置为输出引脚。D 口引脚 DDDn 的作用见表 2．23。

表 2.23　D 口引脚的 DDDn 作用

DDDn	PORTDn	I/O	上　拉	注　　释
0	0	输入	否	三态(高阻)
0	1	输入	是	上拉低 PDn 脚输出电流
1	0	输出	否	推挽 0 输出
1	1	输出	否	推挽 1 输出

n:7,6,…,0 为引脚数。

三、D 口的第二功能

/RD——PORTD，位 7

RD 是外部数据存储器该选通。

/WR——PORTD，位 6

WR 是外部数据存储器写选通。

OC1——PORTD，位 5

OC1 表示比较匹配的输出。PD5 可以作为定时器/计数器 1 的比较匹配的外部输出。为了实现该功能，PD5 应被配置为输出（DDD5 设置为 1）。详细说明和怎样触发该输出请见定时器/计数器 1 的描述。OC1 引脚还可以作为 PWM 模式下定时功能的输出。

INT1——PORTD，位 3

INT1 为外部中断源 1。PD3 引脚可作为 MCU 的外部中断源，详见中断部分。

INT0——PORTD，位 2

INT0 为外部中断 0。PD2 引脚可作为 MCU 的外部中断源，详见中断部分。

TXD——PORTD，位 1

发送数据（UART 的数据输出引脚），当 UART 数据输出允许时，该引脚被作为输出而不管 DDRD1 的值。

RXD——PORTD，位 0

接受数据（UART 的数据输入引脚），当 UART 数据输入允许时，该引脚被作为输出而不管 DDRD0 的值。当 UART 强制该引脚为输入时，PORTD0 的一个逻辑 1 将开房内部的上拉。

四、D 口原理图

D 口原理图如图 2.43~2.49 所示，注意所有的端口引脚都是同步的，图中未画出同步锁存器。

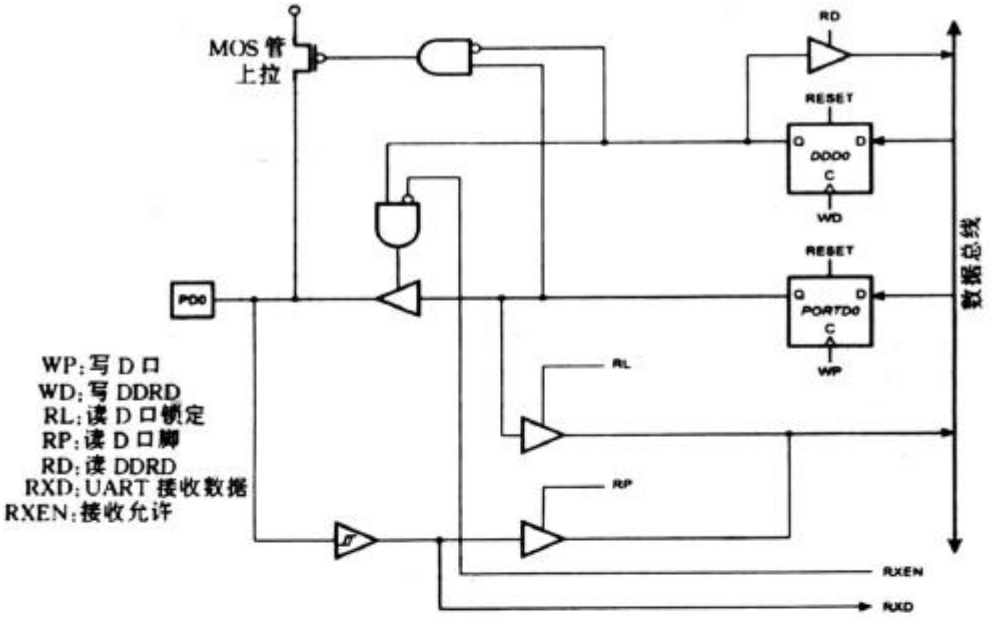


图 2.43　D 口原理图(PD0 脚)

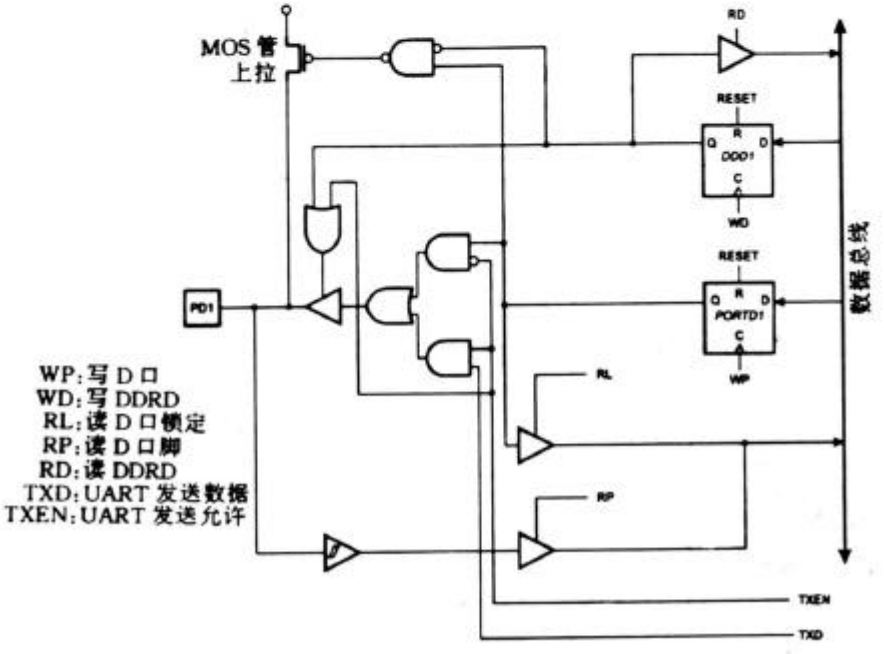


图 2.44　D 口原理图(PD1 脚)

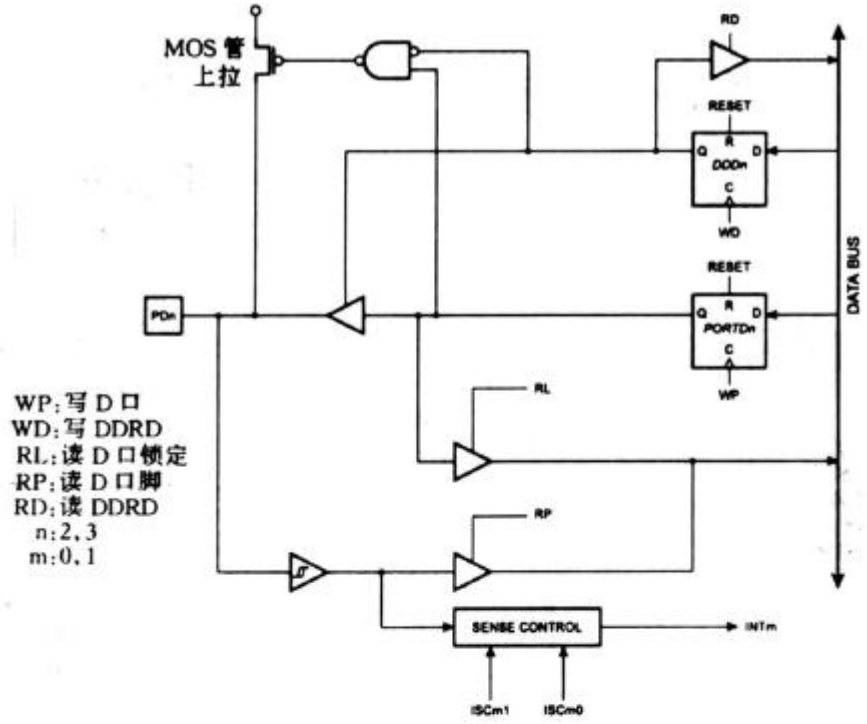


图 2.45　D 口原理图(PD2 和 PD3 脚)

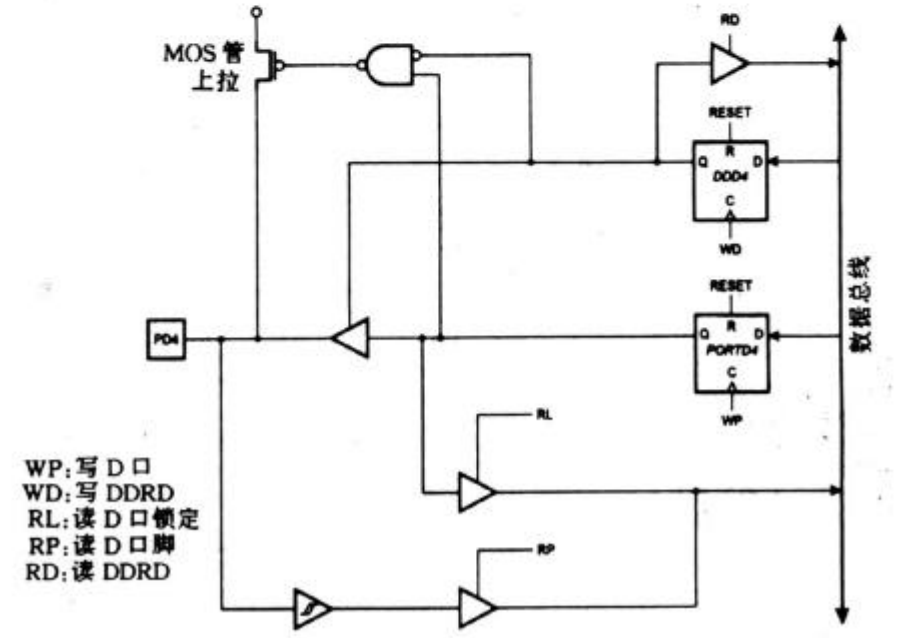


图 2.46　D 口原理图(PD4 脚)

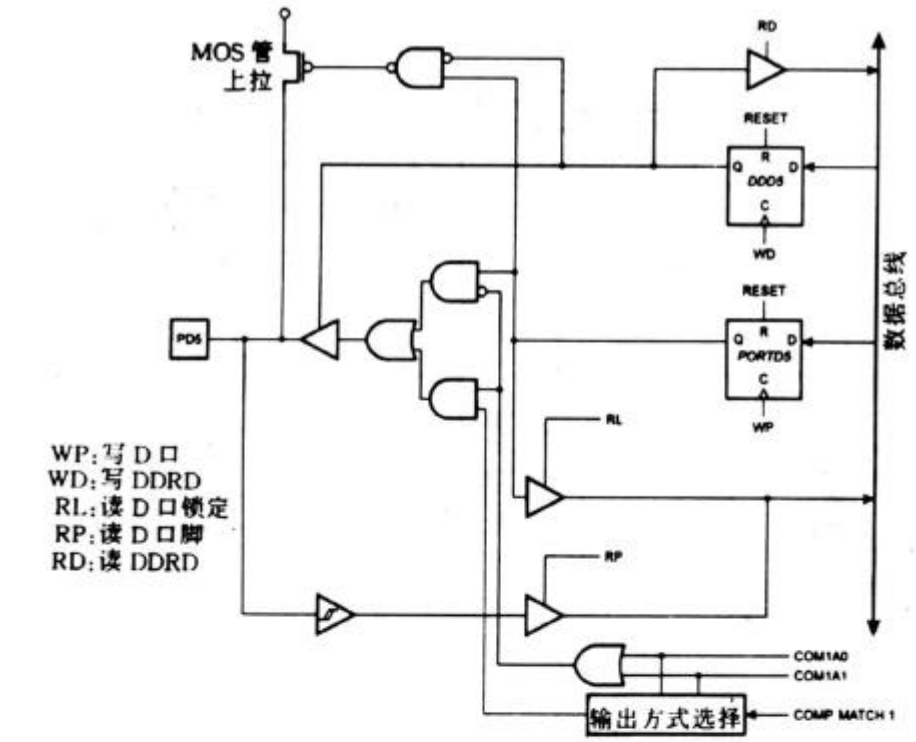


图 2.47　D 口原理图(PD5 脚)

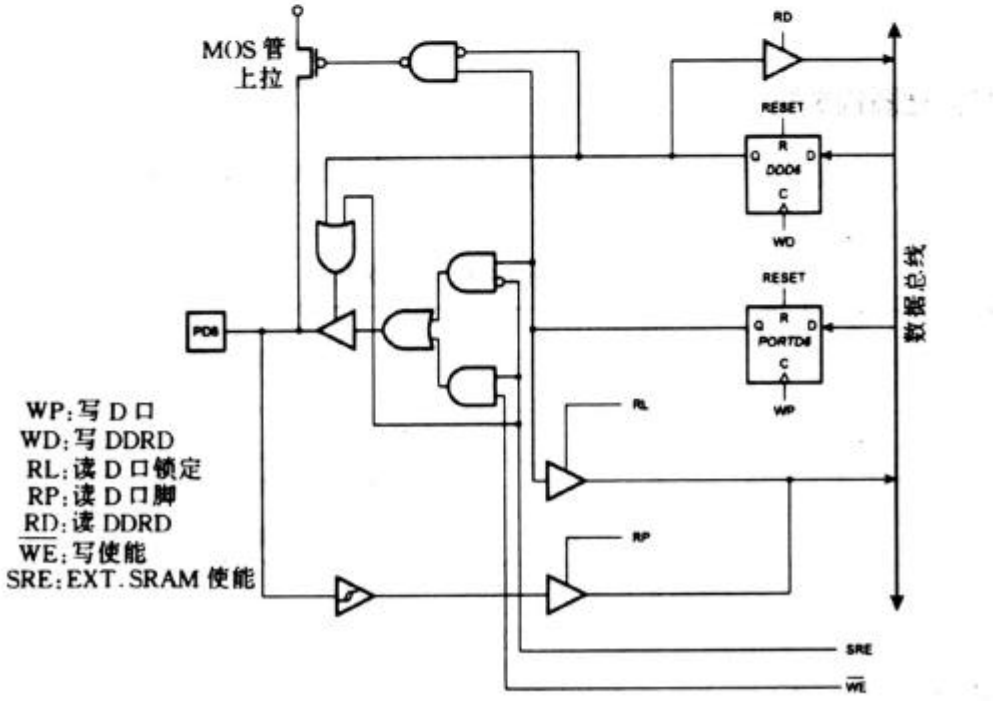


图 2.48　D 口原理图(PD6 脚)

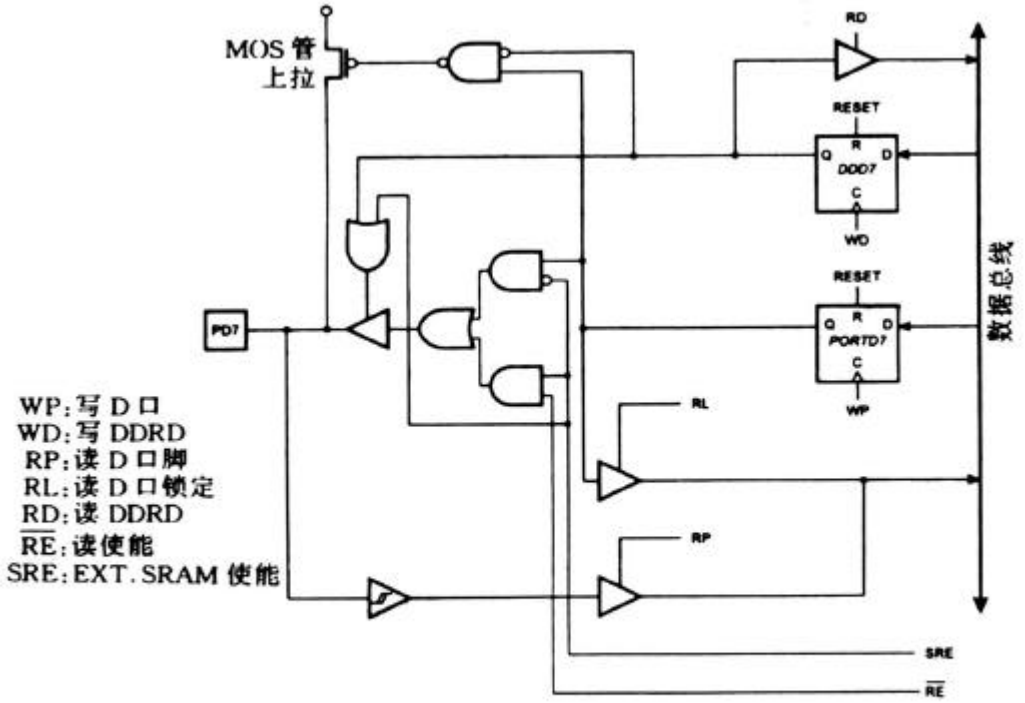


图 2.49　D 口原理图(PD7 脚)

2.12　AVR 单片机存储器编程

2.12.1　编程存储器锁定位

90 系列单片机 MCU 提供 2 个加密锁定位，可以不编程(1)或编程为(0)，而获得表 2.24 的额外特性。

表 2.24　锁定位保护方式

编程锁定位			保护类型
模式	LB1	LB2	
1	1	1	无编程锁定特征
2	0	1	Flash 的再编程被禁止
3	0	0	同 2 模式，但校验被禁止

注意：锁定位只能整片擦除时才能擦除。

2.12.2　熔断位

90 系列单片机有两个熔断位，SPIEN 和 FSTRT

当 SPIEN 被编程为 0 时，串行编程下载被允许，缺省值是被编程的(0)。

当 FSTRT 被编程为 0 时，选择短的启动时间，缺省值为擦除(1)，定货时可以要求该位被编程。

这些位不能被串行编程模式访问，也不能被全片擦除所改变。

2.12.3　芯片代码

所有的 ATMEL 微控制器都有 3 字节的电子标签指定该器件的型号，该标签可以被串行模式和并行模式读出，这三个字节属于不同的地址空间，对于 90 系列单片机它们是：

（1）$000：$1E（指出厂商是 ATMEL）。

（2）$001：$93（指出 4K 字节的 Flash 存储器）。

（3）$002：$01（当$001 是$93 时指出是 AT90S8515 器件）。

2.12.4　编程 Flash 和 E2PROM

90 系列单片机提供了 8K 字节的系统在线可编程的 Flash 程序存储器和 512 字节的 E2PROM 数据存储器。

90 系列单片机通常被售出时，片内的 FLASH 程序存储器和 E2PROM 数据存储器阵列是被擦除的状态（即内容=$FF），而可以被编程。该器件支持高压的（12V）并行编程模式和低压的串行编程模式，＋12V 电源仅用于编程触发，并不从该引脚获取意义上的电流，串行模式提供了对 90 系列单片机方便的在线程序和数据下载方式。

在两种模式下 90 系列单片机的程序和数据存储器阵列是按字节编程的。对于 E2PROM，在串行编程模式中提供了一个自动擦除周期。

2.12.5　并行编程

该节描述了怎样并行编程和验证 90 系列单片机的 Flash 程序存储器、E2PROM 数据存储器及程序存储器的锁定位和熔断位。AT90S8515 的并行编程如图 2．50 所示。

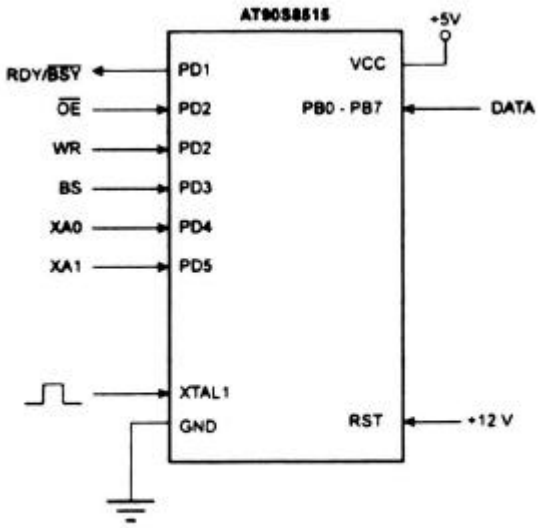


图 2.50　并行编程

一、信号名称

本节中 90 系列单片机的一些引脚被定义为描述并行编程的信号名称，表 2．25 中没有描述的引脚仍用引脚名来表示。

XA1／XA0 位决定了当 XTAL1 引脚结出一个正脉冲时的动作，这两位的设置见表 2．26。

当脉冲是/WR 或/OE 时，装入的命令决定了输入或输出的行为，该命令是一个字节，每一位的功能见表 2．27。

表 2.25　引脚名映射

编程方式的信号名	引脚名	I/O	功　　能
RDY/$\overline{BSY}$	PD1	O	0:器件正在编程,1:器件就绪等待新的命令
$\overline{OE}$	PD2	I	输出允许(低激活)
$\overline{WR}$	PD3	I	写脉冲(低激活)
BS	PD4	I	字节选择
XA0	PD5	I	XTAL 用于位 0
XA1	PD6	I	XTAL 用于位 1

表 2.26　XA1 和 XA0 编码

XA1	XA0	当 FLASH 被 PULSED 时操作
0	0	装入 Flash 或 EEPROM 地址(由 BS 确定的 Fsash 高或低地址字节)
0	1	装入数据(由 BS 线确定的 Fsash 高或低数据字节)
1	0	装入命令
1	1	闲置

表 2.27　命令字节位编码

位号	当 设 置 时 的 定 义
7	芯片擦除
6	写熔断位,在数据字节中定位下列位位置:D5:SPIEN 熔断丝,D0:FSTRT 熔断丝(注意:写 "0"为编程,写 "1"为擦除)
5	写锁定位,在数据字节中定位下列位位置:D7:LB1,D6:LB2,D5:SPIEN 熔断,D0:FSTRT 熔断(注意: "0"为编程)
4	写 Flash 或 EEPROM 中读(由位 0 测定)
3	读标记行
2	读锁定位和熔断位,在数据字节中定位下列位位置:D1:LB1,D0:LB2(注意:写"0"为编程)
1	从 Flash 或 EEPROM 中读(由位 0 测定)
0	0:Flash 访问,1: EEPROM 访问

二、进入编程模式

下列算法使器件进入并行编程模式:

(1)在 Vcc 和 GND 之间加上 4. 5~5. 5 V 电压。

(2)把/RESET 和 BS 设置为 0，并等待至少 100 ns。

(3)把/RESET 加到 12V 且在改变 BS 之前至少等待 100ns。

三、全片擦除

全片的擦除功能将擦除 Flash 和 E2PROM 存储器和锁定位。锁定位在程序存储器被完全擦除之前不会被清除，熔断位并不改变。在编程之前必须执行一个全片擦除。

装载“全片擦除”命令:

（1）把 XA1 和 XA0 设置为（10），触发命令装入。

（2）把 BS 设置为 0。

（3）设置 PB（7~0）为“1000 0000”，这是全片擦除命令。

（4）给 XTAL1 一个正脉冲，这将装人命令并且开始擦除 Flash 和 E2PROM 阵列。在 XTAL1 脉冲之后，给/WR 一个负脉冲使得锁定位在擦除周期结束时被擦除。然后等待 10ms 使擦除完成。全片擦除不生成 RDY / /BSY 信号。

四、编程 Flash

装载“编程 Flash”命令：

（1）设置 XA1、XA0 为“10”，触发命令装入。

（ 2）把 BS 设为 0。

（3）把 PB（7～0）设为“0001 0000”，这是 Flash 编程命令。

（4）给 XTAL1 一个正脉冲，这将装入该命令。

装入地址低字节

（1）设置 XA1、XA0 为“00”，触发地址装入。

（2）设置 BS 为 0，选择低位地址。

（3）设置 PB（7～0）=低位地址字节（$00～$FF）。

（4）给 XTAL1 一个正脉冲，这将装入地址低字节。

装入地址高字节

(1)设置 XA1、XA0 为“00”，触发地址装入。

(2)设置 BS 为 1，选择高位地址。

(3)设置 PB（7～0）=高位地址字节（$00～$0F）。

(4）给 XTAL1 一个正脉冲，这将装入地址高字节。

装入数据字节

(1)设置 XA1，XA0 为“01”，触发数据装入。

（2）设置 PB（7～0）=数据低字节（$00～$FF）。

（3）给 XTAL1 一个正脉冲，这将装入数据低字节。

写入数据低字节

(1)设置 BS=0，选择数据低字节。

(2)给/WR 一个负脉冲，开始编程数据低字节，RDY／/BSY 为低电平。

(3)等待直到 RDY／/BSY 变高时再编程另一个字节。

装入数据字节

(1)设置 XA1、XA0 为“01”，触发数据装入。

(2)设置 PB（7～0）=数据高字节（$00～$FF）。

(3)给 XTAL1 一个正脉冲，这将装入数据高字节。

写入数据高字节

（1）设置 BS＝1，选择数据高字节。

（2）给/WR 一个负脉冲，开始编程数据低字节，RDY／/BSY 为低电平。

（3）等待直到 RDY／/BSY 变高时再编程另一个字节。

装入的命令和地址在编程过程中保持不变，为了简化编程，请考虑如下：

- 编程 Flash 存储器的命令仅在编程第一个字节时需要被装入。
- 地址高字节仅在编程 Flash 中一个新的 256 字节页之前需要装入。

图 2．51 和图 2．52 为可编程 Flash 低字节和可编程 FLASH 高字节。

五、编程 E2PROM

编程 E2PROM 的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（1）装人命令“0001 0001”。

（2）装入低位 E2PROM 地址（$00～$FF）。

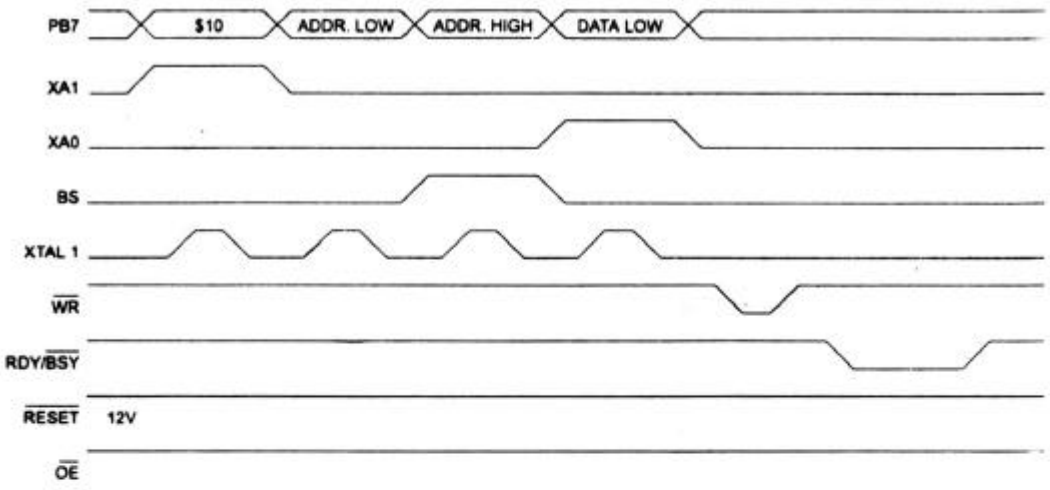


图 2.51　可编程 Flash 低字节

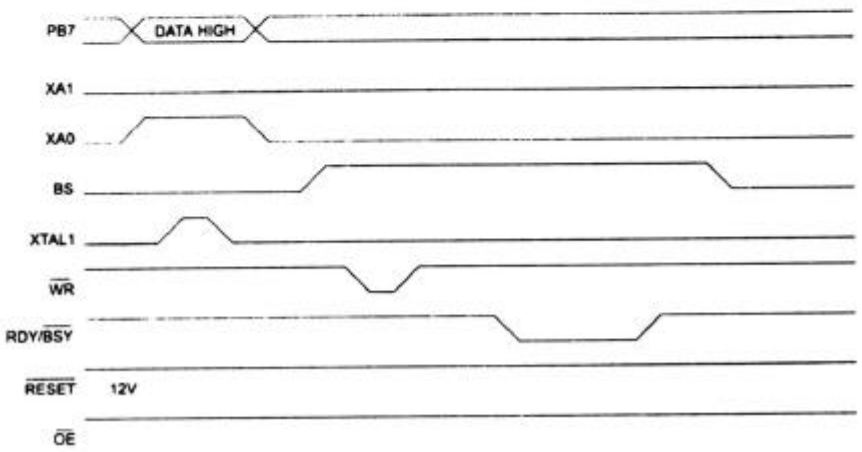


图 2.52　可编程 Flash 高字节

（3）装入高位 E2PROM 地址（$00～$01）。

（4）装入低位 E2PROM 数据（$00～$FF）。

（5）给/WR 一个负脉冲并等待 RDY / BSY 变为高电平。

仅在编程第一个字节之前需要装入命令。

六、读 Flash

读 Flash 的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（1）装人命令“0000 0010”。

（2）装入低地址（$00～$FF）。

（3）装入高地址（$00～$0F）。

（4）设置/OE 为 0，BS 为 0，这时数据低字节可从 PB（7～0）读出。

(5)设置 BS 为 1，这时数据高字节可从 PB（7～0）读出。

（6）设置/OE 为 1。

仅在读第一个字节之前需要装入命令。

七、读 E2PROM

读 E2PROM 的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（1）装人命令“0000 0011”。

（2）装入低地址（$00～$FF）。

（3）装入高地址（$00～$01）。

（4）设置/OE 为 0，BS 为 0，这时数据低字节可从 PB（7～0）读出。

（5）设置/OE 为 1。

仅在读第一个字节之前需要装人命令。

八、编成熔断位

编程熔断位的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（1）装人命令“0100 0000”。

（2）装人数据：

位 5＝0，编程 SPIEN 熔断位；位 5=1，擦除 SPIEN 熔断位。

位 0＝0，编程 FSTRT 熔断位；位 5＝1，擦除 FSTRT 熔断位。

（3）给/WR 一个负脉冲，然后等待 RDY／/BSY 变高。

注意：/WR 必须保持低至少一个毫秒的电平。

九、编程加密位

编程加密位的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

(1）装人命令“0010 0000”。

（2）装入数据：

位 2＝0，编程加密位 2；

位 l＝0 编程加密位 1。

（3）给/WR 一个负脉冲，然后等待 RDY／/BSY 变高。

加密位仅在全片擦除的时候被清除。

十、读熔断和加密位

读熔断和加密位的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（l）装人命令“0000 0100’。

（2）设置/OE 为 0，BS 为 1，这时熔断和加密位的状态可从 PB（7～0）读出。

位 7：加密位 1（0 表示已被编程）。

位 6：加密位 2（0 表示已被编程）。

位 5：SPIEN 熔断位(0 表示已被编程，1 表示被擦除）。

位 0：FSTRT 熔断位（0 表示已被编程，1 表示被擦除）。

（3）设置/OE 为 1。

特别注意 BS 需要设置为 1。

十一、读电子标签字节

读电子标签字节的算法如下（参考 Flash 编程部分的命令、地址和数据的装载）：

（1）装人命令“0000 1000”。

（2）装入低位地址（$00～$02）。

（3）设置/OE 为 0，BS 为 0，从 PB（7～0）可以读出选中的标签字节。

（4）设置/OE 为 1。

仅在读第一个字节之前需要装人命令。

2.12.6　串行下载

一、串行下载

在/RESET 接地时，所有的程序和数据存储器阵列都可以由串行 SPI 总线来编程。该串行接口包括引脚 SCK、MOSI（输入）、MISO（输出）。当/RESET 设为低电平后，应先执行编程允许指令，再执行编程／擦除操作。

当编程 E2PROM 时，内部定时编程操作中包含了自动擦除周期（仅仅在串行编程模式下）而无须先执行全片擦除指令。全片擦除指令把程序和数据存储器阵列的每一地址都变成＄FF。

程序和 E2PROM 存储器阵列的地址空间是分开的，程序存储器为＄0000～＄0FFF，而 E2PROM 存储器为＄0000～＄01FF。

可以用通过 XTAL1 提供的外部时钟，也可以在 XTAL1 和 XTAL2 之间加上一个晶振，串行时钟的（SCK）低电平和高电平的最小时间定义如下：

LOW：大于 1 个 XTAL1 时钟周期。

High：大于 4 个 XTAL1 时钟周期。

二、串行编程算法

以串行的方式编程和校验 90 系列单片机，可用以下的算法（见表 2．28 的 4 字节指令格式）：

(1)上电过程：在 VCC 和 GND 之间上电，同时 RESET 和 SCK 设置为 0。（如果编程器不能保证 SCK 在上电期间为低电平，则在 SCK 为低电平 RESET 必须给出一个正脉冲）。如果晶振没有被连到 XTAL1 和 XTAL2 上，则在 XTAL1 上加上 0 到 20MHZ 的时钟。

（2）等待至少 20MS，向 MOSI／PB5 送串行编程触发指令来触发串行编程，请参照上一节的串行时钟输入 SCK 的低电平和高电平的最小时间。

(3)如果执行了全片擦除（在擦除 Flash 时必须执行），等待 10MS，给/RESET 一个正脉冲然后再从第（2）步开始。

（4）通过在相应的写指令中一起提供地址和数据，可以一次把一个字节写入 FLASH 和 E2PROM 阵列中，E2PROM 存储器的每一地址在新数据写入之前被自动擦除。下一个字节 4MS 后再写入。

（5）任何存储器地址都可以通过读指令来校验，该读指令从串行输出 MISO/PB6 返回选中地址的内容。

（6）在编程结束时，/RESET 可以设置为高电平来开始正常操作。

（7）下电过程（如果需要的话）：设置 XTAL1 为 0（如果未用晶振的话）；设置/RESET 为 1；把 VCC 断开。

表 2.28　串行可编程指令设置

指　令	指　令　格　式				操　作
	Byte1	Byte2	Byte3	Byte4	
编程允许	1010　1100	0101　0011	xxxx　xxxx	xxxx　xxxx	在复位变低后，允许串行编程
芯片擦除	1010　1100	100x　xxxx	xxxx　xxxx	xxxx　xxxx	芯片擦除存储器阵列为 8K 或 512K 字节
读程序存储器	0010　h000	bbbb　bbbb	bbbb　bbbb	oooo　oooo	从字地址 a:b 处的程序存储器读 h(高或低)数据 o
写程序存储器	0100　h000	xxxx　aaaa	bbbb　bbbb	iiii　iiii	写 h(高或低)数据 i 到从字地址 a:b 处的程序存储器中
读 EEPROM 存储器	1010　0000	xxxx　aaaa	bbbb　bbbb	oooo　oooo	从地址 a:b 处的 EEPROM 中读数据 o
写 EEPEOM 存储器	1100　0000	xxxx　xxx0	bbbb　bbbb	iiii　iiii	写数据 i 到地址 a:b 处的程序存储器中
写锁定位	1010　1100	111x　x21x	xxxx　xxxx	xxxx　xxxx	写锁定位，置位 1，2＝'0'到程序锁定位
读器件代码	0011　0000	xxxx　xxxx	xxxx　xxxb	oooo　oooo	从地址 b 处读器件代码 o

注意：a＝高位地址；b＝低位地址；h＝0 为低字节，h1 为高字节；o＝数据输出；i＝数据输入；x＝任意；1＝加密位 1；2＝加密位 2。

2.12.7　可编程特性

当把串行数据写入 90 系列单片机时，数据在 CLK 的上升沿被输入。

当从 90 系列单片机中读数据时，数据在 CLK 的下降沿输出，见图 2. 53 的解释。图 2. 54 是串行可编程和校验示意图。

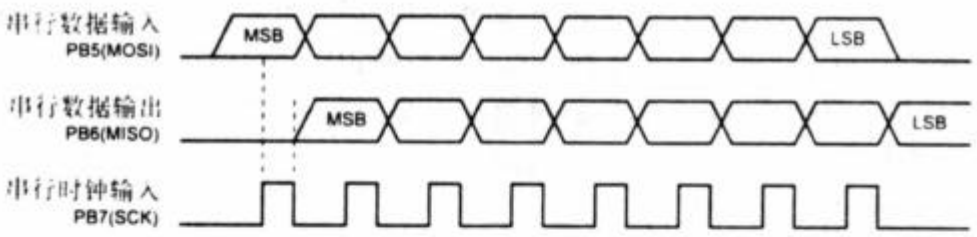


图 2.53　串行下载波形图

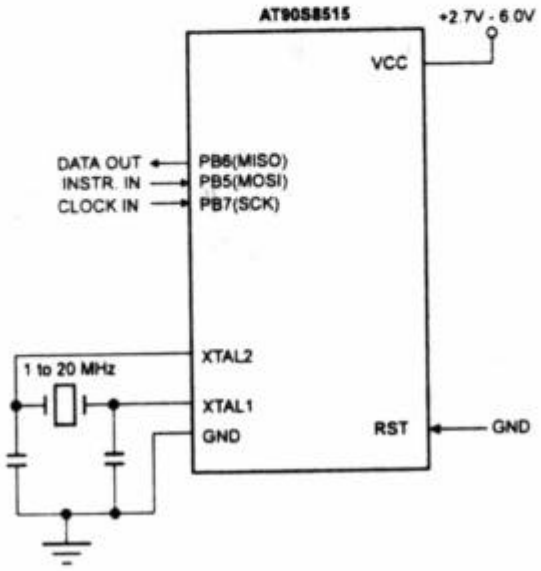


图 2.54　串行可编程和校验