

问:我想使用 AVR 单片机中的 Mega 系列,有哪些开发工具支持这种单片机?

答: AVR 单片机中的 Mega 系列有一套完善的开发工具:

评估/编程板:

ATMEL 的 STK500(980 元/套)起步工具包是适用于开发 Atmega103 芯片的,对芯片的编程是通过在系统编程(ISP)接口完成的。

广州天河双龙电子有限公司的“四合一”SL-AVR 仿真开发实验器(680 元/套);SL-OKAVR 开发实验评估板(380 元/套);SL-AVRL 通用下载编程(ISP)电缆线(180 元/套)。

仿真器:

ATMEL 的 ATICE30 可适用于实时仿真。

C 编译器:

所有的 C 编译器均已在 ATMEL 网站上有关第三方工具供应商的网页上列出。

ATMEL 公司在它的网站:<http://www.atmel.com> 上还提供了许多自由软件,它们可用于对 AVR 单片机的程序进行汇编和模拟,这些软件可以自由下载。

上述产品也可通过 ATMEL 授权的地区分销商和销售代理获得。

IccAVR C 编译器在中国大陆独家代理是广州天河双龙电子有限公司(推广价 1380 元/套),天河双龙电子有限公司还做了很多底层软件模块,建立了库文件 libslavr.a 及头文件 slavr.h,有 LED,LCD,I2C 总线及键扫等模块,IccAVR C 编译器新版本已增加单总线,UART 等模块。

问:有哪些支持 AVR 的起步工具包?

答: STK100,SL-AVR,SL-OKAVR,SL-AVRL 支持 AVR 的 Tiny 系列单片机。

STK200,SL-AVR,SL-OKAVR,SL-AVRL 支持 AVR 单片机,即 AT90S 系列。

STK300,SL-AVR,SL-OKAVR,SL-AVRL 还支持 AVR 的 Mega103 单片机。

STK500,SL-AVR,SL-OKAVR,SL-AVRL 还支持 AVR 的 Mega103 单片机。

所有的起步工具包都包括按钮、LED 指示灯、RS232 通信接口。I/O 口通过分布在 PCB 板边缘的连接插头可以使用。STK200/300 起步工具包还有一条在线编程 (ISP) 电缆,它可用以对 AVR 芯片编程。STK300 还提供了 AVR 应用程序生成器,用来生成外围设备的初始化代码。

SL-AVR 是专门为配合双龙公司主编的《AVR 嵌入式高速单片机原理与应用》一书由北航出版而研制的“四合一”开发实验器。有 6 位 LED 数码管,2*16 的 LCD 显示器,17 键的键盘,通用 DIP 插座,LED 发光二极管,网络电阻,步进电机驱动电路;音响器,配 AT90S8515,有 RS232 通信接口,在线编程 (ISP) 电缆及《双龙 AVR 电子书》光盘,有大量应用实验程序和强大的开发软件。凡购《AVR 嵌入式高速单片机原理与应用》一书(40 元/本),凭本内优惠证购 SL-AVR 开发实验器,可八折优惠(544 元/套);可供选购件:4.5V 二相六线步进电机(30 元/只);5V 开关电源(40 元/只);16X16 点阵汉字显示模块,包括汉字库组成软件及应用例子(90 元/套)。

SL-OKAVR 有通用 DIP 插座,LED 发光二极管作 I/O 状态显示,音响器,配 AT90S1200,有 RS232 通信接口,在线编程 (ISP) 电缆及《双龙 AVR 电子书》光盘,有大量应用实验程序和强大的开发软件。

SL-AVRL 适合 AVR 开发高手及经济条件较差者 AVR 初学者,有 RS232 通信接口,在线编程 (ISP) 电缆及《双龙 AVR 电子书》光盘,有大量应用实验程序和强大的开发软件。

问:即使看门狗没有使用,我也有时在 AVR Studio 调试软件的状态栏中看到“WDT reset”(WDT 复位)的信息,这是为什么?

答:当目标器件运行在较高的频率(大于 6 MHz)时,来自扁平电缆(FPC 电缆)的噪声信号可能错误地触发 WDT 复位检测电路。这对程序的仿真和执行没有影响,它们不会由于噪声的原因而产生复位。噪声仅仅影响 AVR Studio 调试软件的信息处理,在这种情况下不要理睬这个信息。当日

标器件运行在较低的频率（小于或等于 4 MHz）时，应该不会出现这个问题。

问：我刚买了一台 ICE 200 仿真器，在使用不同的芯片和不同的时间运行诊断程序时都是失败的，而其它一切正常，这是什么原因？

答：在 ICE200 仿真器的第一批产品中，是一个固件（监控程序）的问题而导致这个现象，这个现象可以通过升级新版本的监控程序来修正。当使用 2.02 版或更高版本的 AVR Studio 软件时，AVRStudio 将读取固件版本并检查这个问题。如果你的 ICE200 需要升级，它会引导你完成升级处理。这个检查工作只有在装载应用程序时才能被执行，而当运行诊断程序时是不被启动的。如果你首先尝试运行诊断程序诊断仿真器，诊断程序不会检查固件版本并且可能诊断失败。如果你碰巧有这种情况，不要理睬诊断失败而直接去装载应用程序，这时你将可以升级 ICE200 的监控程序。在升级后再回过头来运行诊断程序，诊断结果应该全部是正确的。（注：现在销售的 ICE200 已没有这个问题了）

问：如何检查我的 AVR 仿真器的版本号？

答：在仿真器被连接的时候打开 AVR Studio 程序，选择菜单中 Help->About->Info 来获得你的仿真器的有关信息。AT90ICEPRO 在 1.20 之前的版本和 ATmegaICE 在 1.11 之前的版本是不能被 AVR Studio 软件检测到的。较新的版本已经有了而且可以从 Atmel 网站自由下载。

问：程序在仿真器或单片机中不工作，而在模拟器中看来工作很正常，这是什么原因？

答：这是堆栈指针设置中的常见问题。对所有带 SRAM 的 AVR 单片机，堆栈指针必须设置在 SRAM 最后的地址。

在汇编程序中这样做是正确的：

```
ldi R16, low(RAMEND) ; 装载堆栈指针的低字节到 R16
out SPL, R16          ; 输出这个数值到堆栈指针寄存器
ldi R16, high(RAMEND) ; 装载堆栈指针的高字节到 R16
out SPH, R16         ; 如果 AVR 的 SRAM 小于 256 字节，就将这行注释掉。
```

在 C 程序中，堆栈指针是在启动代码中自动设置的。注意：链接文件(XCL 文件)定义了堆栈指针的位置。请参考应用笔记“AVR032”---- C 编译器的链接文件”。

问：从哪里我能找到完整的 AVR 指令集？

答：指令集在 Atmel 的网站的 Products->AVR 8 bit RISC->Datasheets 下面的网页上，或者直接连接 <http://www.atmel.com/atmel/products/prod200.htm> 网页；<<AVR 嵌入式高速单片机原理与应用>>附录 3 AVR 指令速查表。

问：当开始做一个使用 AVR 单片机的新项目时，最常见、最共同问题有哪些？

答：最常见、最共同的错误是忘记对不带硬件堆栈的芯片设置堆栈指针。对 AT90S8515 单片机上这样做是妥当的：

```
LDI    R16, low(RAMEND)
OUT    SPL, R16
LDI    R16, high(RAMEND)
OUT    SPH, R16
```

注：在 ATMEL 的汇编语言源程序中，经常将 R16 寄存器用 temp 符号来代替。

另一个常见的错误是将端口作为输出使用时，忘记设置端口。在 AVR 单片机中设置端口作为输出使用，是通过写一个“1”到数据方向寄存器(DDR)中完成的。比如设置端口 PORTB 输出高电平：

```
LDI   R16, 0xFF           ; 装载十六进制数 FF 到寄存器 R16
OUT   DDRB, R16          ; 设置端口 PORTB 作为输出使用
OUT   PORTB, R16         ; 设置端口 PORTB 输出高电平
```

使用 AVR 汇编器时第三个常见的错误是：在使用装载程序存储器(LPM)指令时指向了程序存储器中错误的地址。AVR 的程序存储器是组织成字（16 bit）的形式，而 LPM 指令是读取字节（8 bit）。LPM 指令可以读 16 位字的高位字节或低位字节中的任意一个。因为这个原因，必须将 16 位的程序存储器地址分解成两个指向你所希望装入的字节的地址。

这些是在应用笔记 102----“块拷贝例程”中的描述。

问:我在 AT90ICEPRO 中使用回溯 (Trace) 功能选项感觉很困难, 好象每次都不能启动回溯功能, 为什么?

答: 让我们假设你在地址 0x20 处停止仿真, 如果你现在设置启动回溯功能的触发地址是同一个地址即也是 0x20, 那么在程序重新到达 0x20 之前的回溯功能是无效的。原因是触发条件与断点信息应该在指令之前装入 AVR 核中, 所以尽管指令和触发条件在 0x20 地址, 但对在这之前装入 AVR 核中的指令没有任何效果。如果你设置触发条件为一个程序还没有到达的地址, 应该不会有任何问题。

问: Mega103/603 微控制器的哪些引脚用于在线编程(ISP)?

答: Mega103/603 微控制器用于在线编程界面的引脚如下列:

SCK - SCK	(PB1, pin 11)
MISO - PDO	(PE1, pin 3)
MOSI - PDI	(PE0, pin 2)
RESET- RESET	(RESET, pin 20)

PEN(编程使能)引脚和 RESET 引脚可以选择其中一个使用, 这两个引脚有同样的功能(它们都激活内部 iRESET 信号), 但 PEN 引脚只有在上电时(VCC 大约 1.2V)被读取。如果 PEN 引脚不使用, 推荐将此引脚接 VCC。所有其它型号的 AVR 单片机都是使用 RESET 引脚来允许(使能)串行编程的。

问:我不能使用 AVR 仿真器正确地驱动外部负载, 当我使用电压表来测量仿真器 POD 上的引脚时好象引脚不能驱动负载, 这是什么错误?

答: 这是 1.21 版的 ICEPRPO 和 1.10 版的 MegaICE 存在的问题。

可用于仿真器升级的新版本软件可以从 <http://www.atmel.com> 网站的 products->AVR 8bit RISC->software 网页中下载。

为升级 ICEPRPO 的版本到 1.22 版, 请下载 UPGRD122.ZIP 文件。

为升级 MegaICE 的版本到 1.11 版, 请下载 M111UPGR.ZIP 文件。

这些文件是压缩文件, 它包括了一个适用于仿真器的新的程序文件和一个 PC 程序。PC 程序可以用来对代码检查和对仿真器自动升级。

问:我使用了带 A/D 转换器的 AVR 单片机, 单片机工作非常好, 我也喜欢 AVR 的速度。但我有一个问题: 当我改变 A/D 转换器的输入通道时, 好象 A/D 转换器读取的数据仍然是上一个通道的。如果

我第二次读，我又得到正确的数值。为什么会发生这件事？

答：A/D 控制和状态寄存器（ADCSR）中有两个位是用来检测 A/D 转换的结束的：

当一个新的转换启动时 ADSC（启动转换控制位）是用于测试的，不管是否准备就绪，前一次的转换结果仍然被送入 A/D 数据寄存器。当读取 A/D 数据寄存器时，读到的自然仍是上一次的转换结果。

为了得到最近的转换结果，应该测试 A/D 中断标志位（ADIF）或使能全局中断并且在中断服务程序中读取。

例子代码：

```
ldi    R16, 1                ; 选择通道
out    ADMUX, R16
sbi    ADCSR, adif          ; 复位中断标识
sbi    ADCSR, adsc          ; 开始 A/D 转换
wait: sbis   ADCSR, adif     ; 等待直到 ADIF 已经置位
      rjmp   wait
```

问：当将一个文件装入 AVR STUDIO 时，它始终使用我第一次使用过的文件工作，这是什么错误？

答：“AVD”文件包含了有关窗口设置和当前目标文件的信息。有时由 AVR STUDIO 创建的 AVD 文件可能被破坏。根据文件破坏部位的不同，会出现不同的现象。当运行 AVR STUDIO 过程中出现奇怪问题时，首先应尝试删除位于存放源代码的文件夹中扩展名为 avd 的文件。

问：在 AVR 系列中如 AT90S2313 单片机，全部数据存储器不超过 256 字节，而且它不能访问外部存储器。能否忽略 X、Y 和 Z 指针的高位字节，而仅使用指针的低位字节来访问可以使用的存储器？

答：在全部数据存储器（寄存器、I/O 寄存器和 SRAM）少于 256 字节的 AVR 单片机中，当使用 LD/ST 指令访问数据存储器时，CPU 将忽略指针的高位字节。这时允许每个指针的高位字节（R27、R29 和 R31）用于普通的储存用途。自增和预减指令（如 LD -Z, Rd）不会影响高位字节。

注意：当使用 Z 指针访问 FLASH 程序存储器和使用 ADIW/SBIW 指令时，将影响使用的 Z 指针的高位字节。

问：在 IAR C 编译器中怎样才能访问 AVR 内部的 EEPROM 数据存储器？

答：AVR 内部的 EEPROM 是 AVR 微控制器 I/O 存储器的一部分（以 I/O 接口形式访问），不能使用普通变量进行访问。IAR 在标准库例程“ina90.h”中定义了特殊的宏来读写 EEPROM。

读写 EEPROM 的宏有下列原形：

```
_EEGET(VAR, ADR) /* 从 EEPROM 的地址 ADR 读取数值 */
_EEPUT(ADR, VAL) /* 写 VAL 到 EEPROM 的地址 ADR */
```

IAR C 编译器不支持将常量定义到 EEPROM 中（不能生成 EEPROM 的初始化数据文件）。

用 AVR 汇编器来做这件事是很容易的，先写一些定义 EEPROM 中常量的代码：

```
.ESEG
.org 0x50
datatable: .db $01,$02,$03,$04,$05
dtableend: .db $00
```

当这段代码被汇编后，它产生一个 .EEP 文件。这个文件可以被下载到微控制器（MCU）或仿真器中。这个 .org 声明给出数据的地址。这个变量在 C 编译器中可以用 _EEGET() 和 _EEPUT() 进行访问。

问:我无法使MCU在掉电模式时消耗的电流下降到数据簿中的值,我应该怎么做?

答:当AVR进入掉电模式时,I/O引脚的状态应该是不变化的。如果你有一些未连接引脚是三态(输入并且没有上拉),这些引脚的电平是浮动的并且很可能引起振荡,从而导致额外的电流消耗。

为解决这种问题,应该激活所有未连接的输入引脚的上拉电阻,这会使这些引脚具有一个明确的逻辑电平。在由于引脚振荡而产生这个问题的例子中,在活动模式(非休眠状态)打开未使用引脚的上拉是一个好方法。

其次,你再查看一下连接到输出引脚的接线。任何输出引脚上的负载都会在掉电模式时消耗额外的电流。

问:我观察到我现有的微控制器在上电过程中I/O端口是活动的。在我的应用中有一个问题:系统一旦设定I/O引脚中的一个为高电平,MCU可以关闭电源。而即使用一个电阻把该引脚电平拉低,在上电过程中它有时仍然出现高电平。因此,当电源打开时系统立刻关闭,在使用者看来系统不能工作。如果我换成AVR,我猜想会有一些同样的问题?

答:大多数微控制器是同步复位,比如在RESET尚未完全完成时MCU需要一定数量的有效时钟。在上电过程中,当VCC上升到启动晶体振荡器的数值之前,上电复位是有效的,但不能影响I/O引脚的数值,而且I/O引脚是处于随机状态。AVR的I/O端口与复位是异步的,而且保证在整个上电过程中是处于三态,换成AVR单片机将可以解决你的问题。

问:怎样处理AVR微控制器的NC引脚(空引脚)?

答:不要将任何线路连接到NC引脚,NC引脚是为将来的需用而保留的。

问:在AVR汇编器中怎样定义字符串常量?

答:字符串可以作为常量定义在FLASH存储器或EEPROM数据存储器中。

例如:在FLASH中定义一个字符串常量:

```
.CSEG
```

```
fstring: .db "This is a string in flash", 0x00
```

在EEPROM中规定一串常量:

```
.ESEG
```

```
eestring: .db "This is a string in EEPROM", 0x00
```

问:我正在编写一个很大的汇编程序,当我往程序中加入新的行时,文件末尾的字符好象消失了。我怎样才能防止这种现象?

答:汇编器的编辑器有30K字节的文件尺寸限制。如果要克服这个限制,文件必须被分成几个模块,并用“.include”链接。还有一种方法,使用其它编辑器(如NotePad、UltraEdit程序等)和命令行方式工作的汇编器(DOS版本)。

问:AT90S8515有8KB的程序存储器。由于RJMP和RCALL指令只能在2K的相对范围内跳转,而这个部件的指令集没有JMP或CALL指令,我怎样做才能跳转到全部的程序存储器中?

答:程序存储器是组织成4Kx16的形式的,所以只有4K的程序存储器地址空间。在汇编器中选择“Options >> Wrap Relative Jumps”选项,这时将允许你跳过程序存储器的边界。例如,如果你从\$FFE至\$00A做一个相对跳转,程序计数器将被增加12,并且隐藏了程序存储器的边界。这个功能只能

用在 8K 的单片机，4K 的单片机不需要隐藏边界，16K 的单片机需要使用 JUMP 和 CALL 指令。

问:当我使用 UART 时，定时/计数器的任意一个是否都可用于产生波特率?

答:不对，有一个专用的定时器被 UART 用来产生波特率。所有定时/计数器只可以用作普通用途。

问:怎样才能从外部将 AT90S1200 从掉电模式下唤醒?

答:要从掉电模式唤醒 AT90S1200，你必须使能外部 INTO 中断并且是低电平触发中断。当你以掉电模式进入休眠状态时，如果 INTO 引脚上低电平的保持时间大于 16384 个内部 RC 振荡周期时，AT90S1200 将被唤醒。内部 RC 振荡器是用于延时 MCU 的启动，直到 XTAL 振荡器工作稳定。参考 AT90S1200 数据簿，RC 振荡器的频率是受 VCC 电压影响的。

问:当我使用 SBI 和 CBI 指令来设置或清除 I/O 端口的一个信号位时，是否会影响同一端口的其它位?

答:不会，不像多数单片机那样。在 AVR 单片机中允许你 100%安全地操作 I/O 端口的信号位。这也适用于整个端口的操作。有疑问可参考每个 I/O 端口的三个地址。

问:为什么每个 I/O 端口有三个地址?

答:为使你能建立 100%安全的系统，AVR 支持真正的读-修改-写 I/O 端口。如果你希望读取 I/O 引脚的物理电平，读 PIN 寄存器。当你希望改变输出时，读 PORT 锁存能确保正确的数据写回到端口保证全部输出。这种方式始终能给出你所希望的结果而不依赖于引脚的物理电平。这个特点使你省去了为建立一个安全的系统而拷贝你的端口数据进入存贮器的所有工作(这个工作使用了许多指令)。当你使用 SBI 和 CBI 指令来设置/清除 I/O 端口的信号位时，必须始终使用 PORT 的地址。

问:为什么 SBI 和 CBI 指令只能对 \$00 - \$1F 的 I/O 寄存器进行操作?

答:所有 AVR 指令中，除少数外都为两个字节长度。这也意味着只有 65,536(64K)种可能的组合来安排指令集。当我们指定 AVR 指令集时，采取一些折衷的办法是为了尽可能充分地利用这 64K 种组合。不像 CISC 微控制器指令可以是一个、两个、三个或更多个字节，在 AVR 结构上我们不能实现全部指令都象这样。作为例子，在一个立即寻址而且包含全部 32 个寄存器地址的指令中，常数需要 8 位而寄存器地址需要另外的 5 位。这种指令将占用全部指令组合空间中的 8K。换句话说，我们只能安排八个这样的指令，而没有更多的指令能够实现。如果做一个 17 位长度的指令，是不经济和不便的方案，不会被考虑。

在设计 AVR 指令集的过程中设置我们听取了许多建议，我们请教了 C 编译器专家，他们在如何调整指令集来适应 C 编译器方面提出了许多意见。作为例子，编译器专家建议我们为 SBCI (带进位的立即数减法)牺牲 ADDI 指令。

对那些缺少的指令，操作也是方便的。AVR 的代码效率应该能证明我们已经找到一种在实现的指令和省略的指令之间进行折衷的好办法。

问:为什么立即寻址指令在寄存器 R0-R15 中不能工作?

答:所有 AVR 指令中，除少数外都为两个字节长度。这也意味着只有 65,536(64K)种可能的组合来安排指令集。当我们指定 AVR 指令集时，采取一些折衷的办法是为了尽可能充分地利用这 64K 种组合。不像 CISC 微控制器指令可以是一个、两个、三个或更多个字节，在 AVR 结构上我们不能实现全部指令都象这样。作为例子，在一个立即寻址而且包含全部 32 个寄存器地址的指令中，常数需要 8 位而寄存器地址需要另外的 5 位。这种指令将占用全部指令组合空间中的 8K。换句话说，我们只能安排八个这样的指令，而没有更多的指令能够实现。而做一个 17 位长度的指令，是不经济和不便的方案，不会被考虑。

在设计 AVR 指令集的过程中设置我们听取了许多建议，我们请教了 C 编译器专家，他们在如何调整指令集来适应 C 编译器方面提出了许多意见。作为例子，编译器专家建议我们为 SBCI（带进位的立即数减法）牺牲 ADDI 指令。

对那些缺少的指令，操作也是方便的。AVR 的代码效率应该能证明我们已经找到一种在实现的指令和省略的指令之间进行折衷的好办法。

问: 为什么没有 EORI 指令?

答: 所有 AVR 指令中，除少数外都为两个字节长度。这也意味着只有 65,536(64K)种可能的组合来安排指令集。当我们指定 AVR 指令集时，采取一些折衷的办法是为了尽可能充分地利用这 64K 种组合。不像 CISC 微控制器指令可以是一个、两个、三个或更多个字节，在 AVR 结构上我们不能实现全部指令都象这样。作为例子，在一个立即寻址而且包含全部 32 个寄存器地址的指令中，常数需要 8 位而寄存器地址需要另外的 5 位。这种指令将占用全部指令组合空间中的 8K。换句话说，我们只能安排八个这样的指令，而没有更多的指令能够实现。而做一个 17 位长度的指令，是不经济和不便的方案，不会被考虑。

在设计 AVR 指令集的过程中设置我们听取了许多建议，我们请教了 C 编译器专家，他们在如何调整指令集来适应 C 编译器方面提出了许多意见。作为例子，编译器专家建议我们为 SBCI（带进位的立即数减法）牺牲 ADDI 指令。

对那些缺少的指令，操作也是方便的。AVR 的代码效率应该能证明我们已经找到一种在实现的指令和省略的指令之间进行折衷的好办法。

问: 我用 "sbr r30,3" 指令在寄存器 30 中设置 D3 位, 好象对 D3 位不起作用, 但 D0 位与 D1 位可以被设置, 我的做法在哪里有错误?

答: “SBR” 可以用来在一个寄存器中同时设置多位(象“CBR”可以用来同时清除寄存器中的多位)。指令的第二个参数不是指定位号，而是用来与寄存器中的数作逻辑或运算。在这个例子中，\$03 与 R30 的内容相或，所以导致最低的两位置位。为了只设置 D3 位，应尝试下列方法的一种：

```
sbr r30, $08
sbr r30, 0b00001000
sbr r30, (1<<3)
```

上面三行都会产生同样的结果，你可以选择一个你最习惯的用法。

问: 在我的多任务系统中, 除了中断标志以外我还需要定义可以设置和清除的标志, 我应该怎样做?

答: 每个标志使用一个寄存器。为了设置标志用：

```
ldi flag, 01
```

或者，如果你想使用低位的寄存器(R0-R15)。在复位后先清除它，然后用：

```
inc flag
```

来设置标志。

当检查和清除标志时，这样做：

```
lsr flag ; 移动标志到进位标志位 (标志清除)
brne flag_was_set ; 如果标志被设置就跳转
```

象在你所有的中断例程中一样在退出之前要恢复状态寄存器，这种方式十分安全。

问: 在 AT90S1200 中如何实现常数表功能?

答: 由于 AT90S1200 没有 LPM 指令，比较有效的方法是将一个较小的常数表放置在 EEPROM 中。你可以使用 EEPROM 地址寄存器和 EEPROM 数据寄存器有效地直接访问常数表。

如：

```
.ESEG          ; 定义 EEPROM 段
.db my_var0=$45
.db my_var01=$4c
.db my_var02=$5f
.CSEG          ; 定义通用代码段
```

如果你的表不适合于放置在 EEPROM 中，你可以象下面这样放置在 FLASH 中：

```
.def tp    =r16    ; 表指针（表索引）
.def output=r17    ; 输出表的中常数
```

table:

```
ldi output,$45
cpi tp,01
breq end
ldi output,$4c
cpi tp,02
breq end
ldi output,$5f
cpi tp,03
breq end
```

end:

```
ret
```

为了访问表中的 2 号元素，可以这样做：

```
ldi tp, 2
rcall table
```

当从表格子程序返回时，“output”将包括“\$4c”。

问: 在没有 ADDI 指令的情况下，怎样将一个常数（立即数）和寄存器的内容相加？

答：一个较简单的方法是减去一个负数。下面的代码是将寄存器 R20 中和 5 的相加的例子：

```
subi r20, -5
```

注意：无论如何，在这种情况下进位标志位不能用于检测溢出。如果你已经将常数放入另一个寄存器，你可以使用通用的 ADD 指令。如果常量是 1，你能使用 INC 指令来增加寄存器的值。

16 位的加法是同样的，下列代码是将 r17: r16 和 \$0b3c 相加的例子：

```
subi r16, low(-$0b3c)    ; 减去低位字节
sbci r17, high(-$0b3c)  ; 减去高位字节
```

问: 为实现指令的单周期运行，是否对外部时钟倍频？

答：没有，这个时钟是 MCU 真正使用的，没有使用时钟分频或倍频电路。

问: AVR 的一个指令周期使用了多少个时钟周期？

答：只有一个时钟周期。例如在寄存器中加入两个数，全部执行时间等于在 XTAL1 引脚上的时钟信号正半周加负半周的时间。

问: 为什么 AT90S8515 称 8K 的 MCU？

答：大多数 CISC 微控制器的指令是变长度的。一些指令由单字节组成，其它的指令也可以是四个字节长度。大多数 AVR 指令是两个字节长度（只有很少也是四个字节），所以 CPU 在启动运行时不能每次只读取和解释一个字节。而我们的代码尺寸基准，始终是和字节相比较的。因此，在 80C51 上填满 8K 的一个 C 程序，肯定也适合的 8K 的 AVR。

问: 我从哪里能找到数据簿?

答: 你可以从网站www.atmel.com有选择地下载它们, 你也可从当地分销商和 ATMEL 办事处得到数据簿。你还可以发电子邮件到literature@atmel.com索取。

问: AVR 有哪些超过我现有 MCU 的附加性能?

答: 十分紧凑的代码, 特别是对 C 程序。

4-10 倍的高速度

可编程 FLASH 存储器, 可在系统编程。

芯片内含 EEPROM 存储器, 可在系统编程。

低功耗。