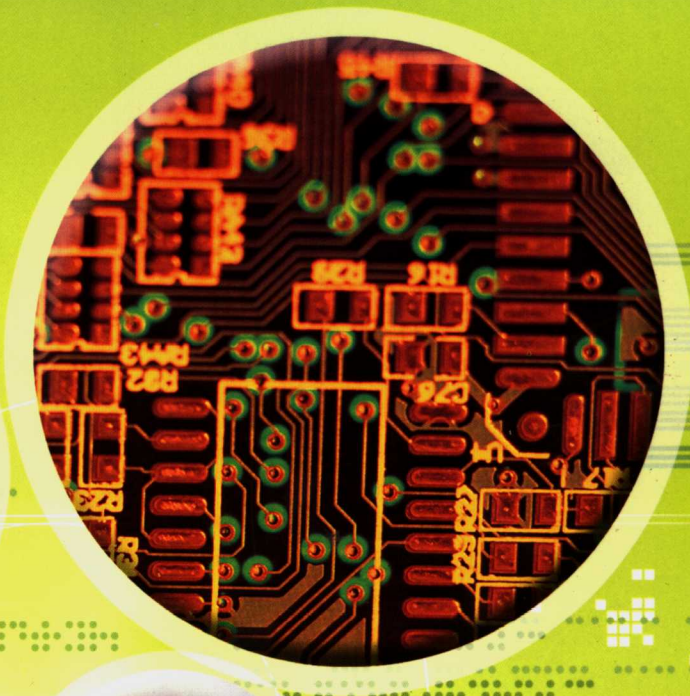
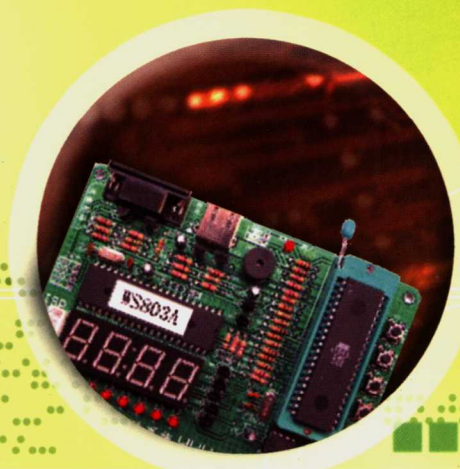


嵌入式 C语言程序设计 ——使用MCS-51

郁文工作室 编著
侯振鹏 改编



 人民邮电出版社
POSTS & TELECOM PRESS

郁文工作室 编著
侯振鹏 改编



图书在版编目 (CIP) 数据

嵌入式 C 语言程序设计: 使用 MCS-51 / 郁文工作室编著; 侯振鹏改编.

—北京: 人民邮电出版社, 2006.7

ISBN 7-115-14855-4

I. 嵌... II. ①郁...②侯... III. ①单片微型计算机, MCS-51—程序设计②C 语言—程序设计 IV. TP368.1②TP312

中国版本图书馆 CIP 数据核字 (2006) 第 062518 号

版 权 声 明

本书中文简体字版由台湾全华科技图书股份有限公司独家授权, 仅限于中国大陆地区出版发行。

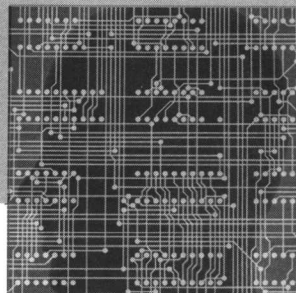
嵌入式 C 语言程序设计——使用 MCS-51

- ◆ 编 著 郁文工作室
改 编 侯振鹏
责任编辑 陈 昇
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 18.5
字数: 446 千字 2006 年 7 月第 1 版
印数: 1—5 000 册 2006 年 7 月北京第 1 次印刷
著作权合同登记 图字 01-2006-3683 号
ISBN 7-115-14855-4/TP • 5454

定价: 34.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

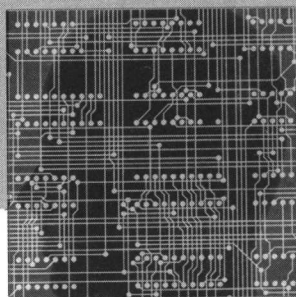
内容提要



本书介绍使用 C 语言编写 8051 单片机程序, 向读者介绍了使用模块化程序设计的技巧, 让编写程序变成一件很容易的事。书中内容包括: MCS51 的硬件体系结构介绍、基本工具的使用、C 语言程序的编写、基本程序范例、高级程序范例及项目开发等。本书还向读者介绍了自己如何制作烧录器。

本书内容由浅入深、循序渐进, 适合于作为大中专学校的单片机课程教材, 也适合于读者自学单片机程序开发。

序



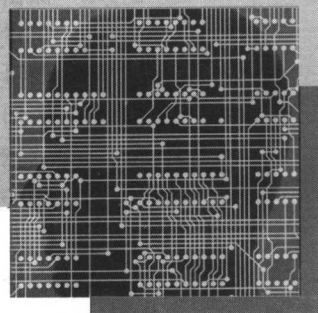
AT89S5X 系列的单片机由 Atmel 公司生产，这一系列单片机与传统的 AT89C51 完全兼容，价格便宜，而且还拥有一些十分优良的特性，因此非常适合于嵌入式系统使用。这些特性包括在线烧录、看门狗等功能。在网上搜索时会发现，此系列芯片的用户遍及全球，这可能与支持 AT89S5X 系列单片机的软件十分丰富、可以自己制作简单的烧录器有关。因为使用 AT89S5X 系列单片机时不会有束手束脚的感觉，所以用户当然喜欢使用。

一个偶然的机会，我使用了这样一个芯片，刚好又在网络上找到了一些资源，所以就自己动手摸索起来，并且开发了一个 AT89S51 串行烧录器，后来又使用 Visual Basic 编写程序烧录 AT89S5X 系列的单片机，这对于经济拮据的学生或研发人员真是一个好消息。于是就想到或许这对于学习单片机的初学者来说是一项不错的选择，因此大胆地将自己的一些经验撰写成书。

本书的程序也适用于 AT89C5X 系统的单片机，我承认这方面高手如云，本人才疏学浅，书中若有疏漏或错误的地方，请诸位高手见谅，并予以指正。

谢谢我的妻儿在本书撰写过程中对我的支持。最后将此书献给我的爸爸，谢谢他对我的教导。

目 录



第 1 章 MCS51 硬件体系结构介绍	1
1-1 89X51 的引脚	3
1-2 89X51 的内存	5
1-3 89X51 的特殊用途寄存器	6
1-4 中断	18
1-5 周边设备的使用	19
1-5-1 外部硬件中断	19
1-5-2 计时计数器	21
1-5-3 串行端口	23
1-6 看门狗	25
1-7 结论	26
第 2 章 基本工具的使用	27
2-1 安装 Keil C 编译器	27
2-2 开始使用 Keil C	32
2-3 软件模拟	40
2-4 自制 AT89S51 的烧录器	43
2-4-1 PonyProg 串行烧录器	43
2-4-2 ISP Flash Microcontroller Programmer	45
2-4-3 本书所提供的串行烧录器	46
2-5 结论	50
第 3 章 编写 C 语言程序	51
3-1 C 语言基础知识	51
3-1-1 编译指令 #include	52
3-1-2 注解	52
3-1-3 基本数据类型	53

3-1-4	用户自定义的数据类型	53
3-1-5	标识符	54
3-1-6	保留字	55
3-1-7	常量	55
3-1-8	变量声明	55
3-1-9	数组	56
3-1-10	运算符	57
3-2	C 语言的控制命令	58
3-2-1	if 语句	58
3-2-2	嵌套 if 语句	59
3-2-3	switch 语句	59
3-2-4	for 循环	60
3-2-5	while 循环	61
3-2-6	do/while 循环	61
3-2-7	标示和 goto 语句	61
3-3	C 语言的指针和函数	61
3-3-1	指针	61
3-3-2	指针与数组	62
3-3-3	指针的运算	62
3-3-4	函数	63
3-3-5	中断服务函数	64
3-4	汇编语言	65
3-5	宏的使用	67
3-6	函数库	68
第 4 章	基本程序范例	75
4-1	LED 的控制	75
4-2	指拨开关的输入	86
4-3	七段显示器的控制	88
4-4	计时器 Timer0 的溢出中断控制	91
4-5	外部中断 INT0	95
4-6	按钮检测 1	99
4-7	按钮检测 2	102
4-8	四个七段显示器的显示控制	106
4-9	4×4 小键盘输入	111
4-10	C 语言程序调用汇编语言程序	118
4-11	结论	134
习题		134
第 5 章	高级程序范例	137

5-1 数字时钟 1	137
5-2 LCD 的显示控制	144
5-3 数字时钟 2	159
5-4 电子音乐	166
5-5 步进马达	175
5-6 D/A 转换器	183
5-7 绘图型显示器	187
5-8 RS232	204
5-9 结论	207
第 6 章 项目开发	209
6-1 密码锁	209
6-2 数字电压表	228
6-3 网络远程控制系统	235
6-4 智能温度计	250
6-5 数字时钟——使用绘图型 LCM	265
6-6 结论	276
附录 A 8051 指令集	279
附录 B 串行烧录的工作原理	283

第 1 章

MCS-51

MCS51 硬件体系结构介绍

本章将介绍 8051 单片机的硬件体系结构。单片机是指在单独一个微处理器中不但包含 CPU，还包含一系列外围设备以及系统所需要的存储器。Intel 的 8051 系列应该是最早的单片机，它的历史大约有 30 年了！在这段时间内，Microchip、AMD、Philips、Mortorola、Hitachi、Toshiba 都追随着 Intel 的步伐，推出了与 8051 兼容或是自己特有的嵌入式微处理器；因此 8051 算是单片机的始祖，它的应用遍及各个领域。

Atmel 公司所生产的 AT89C51，价格便宜，具有内置的闪存，可反复烧录程序，烧录时又很简单，因此使用广泛。目前，Atmel 公司又推出了改良型 AT89S5X 系列单片机，AT89S5X 的引脚和指令跟 AT89C51 单片机完全相同，因此在程序与硬件上完全兼容。AT89C51 在烧录时必须在第 31 支引脚 VPP 提供 +12V 的电压，而且只能够采取并行方式烧录，用户必须购买特殊的烧录器，烧录程序时须将 IC 拔出放到烧录器上，烧录完毕之后，再将 IC 插回实验板或试验电路板。

新型的 AT89S5X 系列单片机，烧录电压只需 +5V，而且可以采取并行或是串行的方式烧录。因为用户可以使用 +5V 的串行烧录方式，因此无论是在实验过程中或是已经完成的产品，都不必拔出 IC。用户只要预留 4 支引脚（SCK、MOSI、MISO、RST）的接头，就可以执行烧录程序存储器的配置工作。这种烧录方式称为在线烧录（In System Programming，简称为 ISP）。而且因为 AT89S5X 系列单片机可以使用 +5V 的串行烧录方式，所以烧录器变得比较简单，用户可以自己制作一台简易型的烧录器。本书第 3 章将介绍如何自制一台 AT89S5X 系列单片机的烧录器，只要花 100~200 元即可完成。

Atmel 公司所推出的 AT89S5X 系列单片机实际上包括了：AT89S51、AT89S52、AT89S53 和 AT89S8252 等。图 1-1 列出了些在线烧录型 89S5X 单片机，请注意它们之间的差别。

图 1-1 中的 AT89S2051、AT89S4051、AT89S8253、AT89LS51、AT89LS52、AT89LS53、AT89LS8252、AT89C51IC2、AT89C51RB2、AT89C51RC2 和 AT89C51RD2 可以在 2.7V 的低

电压下工作。AT89S2051 和 AT89S4051 只有 20 支引脚, I/O 引脚只有 15 支。AT89C5115 有 28 支引脚, I/O 引脚只有 20 支。

AT89S5X 系列的元件	闪存 (KB)	EEPROM (KB)	SRAM (bytes)	工作频率 (MHz)	外部 中断	16 位 计数器	A/D 转换器
AT89S51	4	0	128	0~33	2	2	
AT89S52	8	0	256	0~33	2	3	
AT89S53	12	0	256	0~24	2	3	
AT89S8252	8	2	256	0~24	2	3	
AT89S8253	4	2	256	0~24	2	3	
AT89S2051	2	0	256	0~24	2	2	
AT89S4051	4	0	256	0~24	2	2	
AT89LS51	4	0	128	0~33	2	2	
AT89LS52	8	0	256	0~33	2	3	
AT89LS53	12	0	256	0~24	2	3	
AT89LS8252	8	2	256	0~24	2	3	
AT89C51AC2	32	2	1280	40	2	3	10 位
AT89C51AC3	64	2	2304	60	2	3	10 位
AT89C51ED2	64	2	2048	60	2	3	
AT89C51IC2	32	2	1280	60	2	3	
AT89C51ID2	64	2	2048	60	2	3	
AT89C51RB2	16	2	1280	60	2	3	
AT89C51RC2	32	2	1280	60	2	3	
AT89C51RD2	64	2	2048	60	2	3	
AT89C5115	16	2	512	40	2	2	10 位

图 1-1

以上所列出的 AT89S5X 系列单片机, 它们的基本核心体系结构和传统的 8051 都是相同的, 也就是说 I/O、计时计数器、中断、UART 等周边设备和 AT89C51 单片机是完全兼容的, 因此现有的程序也是兼容的。不过, 这些 89S5X 系列的单片机中有些不容易买到, 或者价格昂贵, 也有一些的包装和引脚不适合用在试验电路板上。AT89S51、AT89S52、AT89S53 和 AT89S8252 的引脚和 AT89C51 的引脚完全相同 (DIP 40), 而且容易买到, 价格也便宜, 因此本书就采用这些单片机。其中 AT89S51 的价格和传统的 AT89C51 差不多, 所以本书就以 AT89S51 为主进行介绍, 但是本书所介绍的内容也适用于其他的 8051 单片机。

AT89S5X 系列的单片机除了提供在线烧录的功能之外, 另外还加入了看门狗计时器 (Watchdog Timer)。AT89S8252 中有 2KB 的 EEPROM, 如图 1-2 是 AT89S5X 系列单片机和 AT89C51 的差别比较。

	在线烧录	EEPROM (KB)	看门狗计时器	工作频率 (MHz)
AT8925X 系列	有	89S8252 才有	有	0~33
AT89C51	没有	没有	没有	24

图 1-2

虽然在本书中, 我们采用的是 Atmel 公司所生产的 AT89S5X 系列的单片机, 但是, 基本上所

有 8051 系列的单片机都具有相同的 8051 核心, 因此我们只要学会其中一种, 其余的都大同小异; 而且它们的指令都是通用的, 也就是说原先为某一种 AT89S5X 微处理器所开发的程序可以直接使用在其他不同公司生产的 8051 单片机上。因此对于初学者而言, 只要先学会其中一种即可。

1-1 89X51 的引脚

图 1-3 是 89X51 的引脚图。

1	P1.0	V_{CC}	40
2	P1.1	P0.0/AD0	39
3	P1.2	P0.1/AD1	38
4	P1.3	P0.2/AD2	37
5	P1.4	P0.3/AD3	36
6	P1.5	P0.4/AD4	35
7	P1.6	P0.5/AD5	34
8	P1.7	P0.6/AD6	33
9	RST	P0.7/AD7	32
10	P3.0/RXD	EA/VPP	31
11	P3.1/TXD	ALE/PROG	30
12	P3.2/INT0	PSEN	29
13	P3.3/INT1	P2.7/A15	28
14	P3.4/T0	P2.6/A14	27
15	P3.5/T1	P2.5/A13	26
16	P3.6/ $\overline{WR}$	P2.4/A12	25
17	P3.7/ $\overline{RD}$	P2.3/A11	24
18	XTAL2	P2.2/A10	23
19	XTAL1	P2.1/A9	22
20	GND	P2.0/A8	21

图 1-3

AT89S5X 的引脚说明如图 1-4 所示。

V_{CC}	电源引脚 +5V
GND	电源接地
端口 0 P0.7...P0.0	端口 0 是一个 8 位的双向 I/O 端口, 内部没有升压电阻, 所以如果使用端口 0 作为输入或输出时, 必须外加升压电阻。在读取外部存储器时, 端口 0 作为地址总线 (A0~A7) 和数据总线 (D0~D7)。 8051 是存取外部存储器时, 端口 0 会先送出低地址 (70~A7), 然后才送出数据或读取数据 (D0~D7), 用户必须使用 ALE 来锁住低地址
端口 1 P1.7...P1.0	端口 1 是一个 8 位双向 I/O 端口, 每一个引脚都有内部升压电阻, 可直接驱动 LED。在存取外部存储器时, 端口 2 还可以用作外部 SRSM 的地址总线的高位 (A8~15)
端口 3 P3.7...P3.0	端口 3 是一个 8 位双向 I/O 端口, 每一个引脚都有内部升压电阻, 可直接驱动 LED
RST	RST 引脚输入超过两个机械周期的高电位, 将引起系统重置
XTAL1	振荡器的输入端
XTAL2	振荡器的输出端
ALE/PROG	ALE 引脚是存取外部存储器时的低位地址锁定信号
EA/VPP	启用外部存储器, 并行烧录内部闪存时, 此引脚加上 +12V 的高电压
PSEN	Program Strobe Enable, AT89S5X 要执行外部存储器的程序时, 此引脚在每一个机械周期当中会产生两个脉冲

图 1-4

晶体振荡器

XTAL1 和 XTAL2 引脚分别是芯片内振荡器的输入和输出端，它可以使用晶体振荡器或是陶瓷振荡器。如果采用外加石英晶体和两个 30pF 的电容器并联以产生工作时钟，其接线如图 1-5 所示。

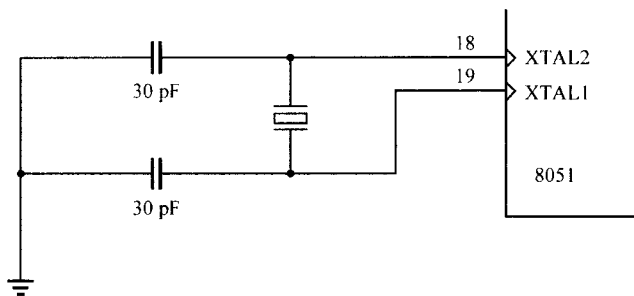


图 1-5

但是用户也可以通过外部的振荡信号产生时钟，当使用外部振荡时钟时，XTAL2 应该空接，如图 1-6 所示。

RST 说明

8051 有 3 种重置方式，下面分别说明。

1. Power-On Reset: 当用户在 8051 的 Vcc 和 GND 之间加上 +5V 的电压时，8051 会被重置。

2. External Reset: 当 8051 的 RESET 引脚上有一个超过两个机械周期的高电位脉冲时，则 8051 会被重置。

3. Watchdog Reset: 当看门狗被启用，而且看门狗计时器在一个看门狗周期完毕之前都没有执行看门狗重置 (watchdog reset) 指令时，则 8051 会被重置。此时 RST 引脚会产生 98 个机械周期的高电位脉冲。

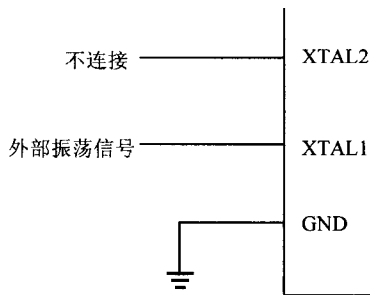


图 1-6

并非所有的 MCS-51 单片机都有看门狗计时器，例如：AT89C51 没有看门狗计时器，而 AT89S51 则有看门狗计时器。

当 8051 被 RESET 之后会进入以下状态：

- (1) 从 \$0000 的位置开始执行程序。
- (2) 所有的中断皆失效。
- (3) 使用寄存器组 0。
- (4) 对于 AT89S5X 单片机而言，看门狗中断无效。

用户可以加上以下的电路作为 CPU 的 RESET 电路。在此电路中，用户按下弹跳按钮之后，8051 会被重置，如图 1-7 所示为 8051 的 RESET 电路。

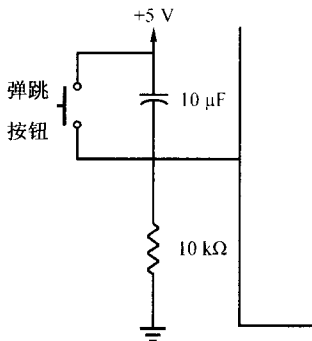


图 1-7

$\overline{EA}$ 引脚说明

当 AT89S5X 读取外部的程序存储器 (0000H~FFFFH) 时，AT89S5X 的第 31 支引脚 $\overline{EA}$

必须接地，此时内部的程序存储器完全失效。

反之，如果 AT89S5X 要读取内部的程序存储器时，第 31 支引脚 $\overline{EA}$ 必须接 +5V。如果用户在烧录 AT89S5X 时，同时也设置烧录锁码位 1 (Lock Bit 1) 时，AT89S5X 会忽略第 31 支引脚 $\overline{EA}$ ，并且直接读取内部的程序存储器。

有些用户在烧录 AT89S5X 时，没有设置烧录锁码位 1，而且把 AT89S5X 接在试验电路板上时也忽略掉第 31 支引脚，以致于第 31 支引脚是空接，但是此时即使第 31 支引脚是空接，也会造成 AT89S5X 只读取外部的程序存储器的问题，因此常常无法执行。所以请特别注意，如果烧录 AT89S5X 时没有设置烧录锁码位，则第 31 支引脚必须接 +5V。

1-2 89X51 的内存

AT89X51 的内部包含了 4KB 的闪存，这 4KB 的闪存是用来烧录程序的存储器；另外，AT89X51 的内部还包含了 128 字节的 RAM，这 128 字节的 RAM 是程序使用的数据存储器，其中有些是 8051 的寄存器组，其余的位置可以用来储存变量，也可以用来当作堆栈区域，如图 1-8 所示。

30H~7FH	一般数据或是堆栈使用的区域
20H~28H	可以针对固定地址的区域
18H~1FH	寄存器组 3 (R0~R7)
10H~17H	寄存器组 2 (R0~R7)
08H~0FH	寄存器组 1 (R0~R7)
00H~07H	寄存器组 0 (R0~R7)

图 1-8

寄存器组

AT89X51 的内部包含 4 个寄存器组，每一个寄存器组中包括 R0、R1、R2、R3、R4、R5、R6 和 R7 等 8 个寄存器。当 AT89X51 刚加上电源或是 RESET 之后，会自动选用寄存器组 0。

用户可以通过设置 PSW 寄存器中的 RS0 和 RS1 位来选择寄存器组，PSW 寄存器的内容如图 1-9 所示。

位	7	6	5	4	3	2	1	0
	CY	AC	F0	RS1	RS0	OV	—	P

图 1-9

用户要选择使用寄存器组 0 时，只要设置 PSW 寄存器的 RS0=0、RS1=0 即可；

用户要选择使用寄存器组 1 时，只要设置 PSW 寄存器的 RS0=1、RS1=0 即可；

用户要选择使用寄存器组 2 时，只要设置 PSW 寄存器的 RS0=0、RS1=1 即可；

用户要选择使用寄存器组 3 时，只要设置 PSW 寄存器的 RS0=1、RS1=1 即可。

如果使用 Keil C 编写您的程序，您只要在程序开头加入 `#include<regx51.h>`，就可以直接在程序中加入 RS0=1 或 RS1=0 等指令，因为文件 regx51.h 中就已经定义好 RS0 和 RS1 等位

的地址值，详细说明请参见下一节。

AT89S52 的内部则包含了 256 字节的 RAM，因此从 30H 到 FFH 的存储器位置都可以同作一般数据区域或是堆栈区域。

堆栈

堆栈 (stack) 是一种数据处理的方式，如图 1-10(a) 所示。微处理器使用数据存储器的一部分作为堆栈区域，并且使用堆栈指针 (Stack Pointer，简称为 SP) 指向堆栈的顶端。当数据存入堆栈时是放在堆栈指针所指到的位置，同时堆栈指针会往上移动 (数值减 1)，如图 1-10(b) 所示，这个动作叫做推入 (push)；当数据从堆栈中读出来时，数据就从堆栈的顶端读出来，同时堆栈指针会往下移动 (数值加 1)，这个动作叫做弹出 (pop)，如图 1-10(c) 所示。因为堆栈存取数据时是由下往上 (由高地址往低地址) 存放，因此一般而言，我们通常把堆栈设置在 RAM 的结尾。编程人员并不需要记住 RAM 结尾的地址值，因为 Keil C 会自动处理堆栈的管理与使用。

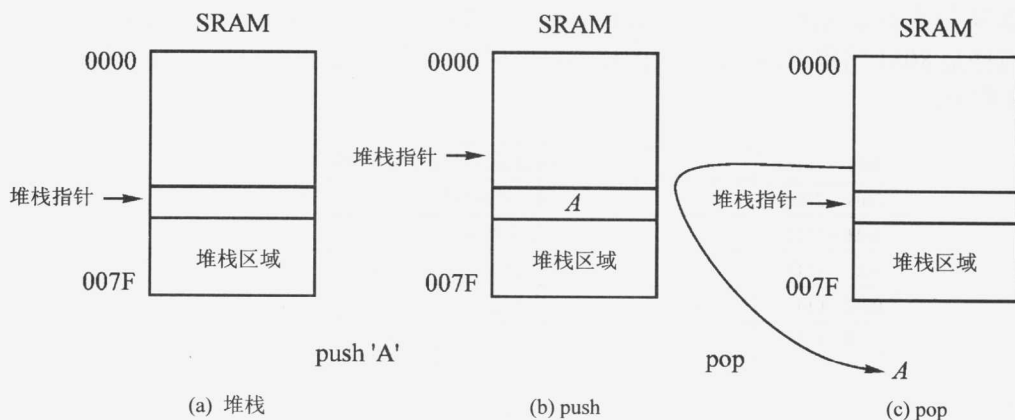


图 1-10

数据指针寄存器

数据指针寄存器 (Data Pointer，简称 DP) 是一个指向存储器位置的 16 位寄存器，数据指针寄存器分为 DPH 和 DPL。DPH 和 DPL 是用来访问存储器内数据的指针。

1-3 89X51 的特殊用途寄存器

AT89X51 内部的数据存储器中，从 80H 到 FFH 的存储器中有些地址由特殊用途寄存器 (Special Purpose Register，简称 SPR) 所使用。这些特殊用途寄存器在 AT89X51 内部的位置如图 1-11 所示。

什么是特殊用途寄存器呢？AT89S5X 微处理器中有许多周边设备，这些周边设备包括了中断控制器、计时计数器、看门狗计时器、串行传输等。使用这些周边设备时必须经由特殊用途寄存器，用户必须设置特殊用途寄存器的各项控制参数，才能控制 AT89S5X 的内部周边设备。因此初学者必须对这些特殊用途寄存器有所了解。以下我们将介绍其中一些特殊用途

寄存器。

	8	9	A	B	C	D	E	F	
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87
	0	1	2	3	4	5	6	7	

图 1-11

1. PSW 寄存器的内容

PSW 寄存器的内容如图 1-12 所示。

位	7	6	5	4	3	2	1	0
	CY	AC	F0	RS1	RS0	OV	—	P

图 1-12

P 位：奇偶位 (Parity Bit)，P=0 表示 A 寄存器中 1 的个数是偶数，P=1 表示 A 寄存器中 1 的个数是奇数。

OV 位：溢出位 (Overflow)，OV=1 表示运算时有溢出产生。

RS0 和 RS1 位：用来选择寄存器组，如图 1-13 所示。

F0 位：用户自行设置的位。

AC 位：辅助进位位，AC=1 表示运算时较低的 4 位有进位产生。

RS1	RS0	选择的寄存器组
0	0	寄存器组 0
0	1	寄存器组 1
1	0	寄存器组 2
1	1	寄存器组 3

图 1-13

CY 位：进位位，CY=1 表示运算时有进位产生。

用户的程序可以通过 PSW 来判断各项工作执行之后的结果，然后采取必要的动作。

2. 中断寄存器 IE 的内容

中断寄存器 IE 用来设置 AT89S5X 是否接受各种中断，当用户要使用各项中断时必须先设置此寄存器。中断寄存器 IE 的内容与说明，如图 1-14 所示。

位	7	6	5	4	3	2	1	0
	EA	—	ET2	ES	ET1	EX1	ET0	EX0

位		
EA	设置整体中断的启动或关闭	EA=0 设置系统不接受所有中断， EA=1 设置系统可以接受中断
ES	设置是否接受 UART 的中断	ES=0 设置不接受 UART 的中断， ES=1 设置可以接受 UART 的中断
ET0	设置是否接受 Timer0 的中断	ET0=0 设置不接受 Timer0 的中断， ET0=1 设置接受 Timer0 的中断
ET1	设置是否接受 Timer1 的中断	ET1=0 设置不接受 Timer1 的中断， ET1=1 是设置接受 Timer1 的中断
ET2	设置是否接受 Timer2 的中断	ET2=0 设置不接受 Timer2 的中断， ET2=1 设置接受 Timer2 的中断
EX0	设置是否接受 INT0 引脚的中断	EX0=0 设置不接受 INT0 引脚的中断， EX0=1 设置接受 INT0 引脚的中断
EX1	设置是否接受 INT1 引脚的中断	EX1=0 设置不接受 INT1 引脚的中断， EX1=1 设置接受 INT1 引脚的中断

图 1-14

在上图中的 ET2 只有 AT89S52 才有，因为 AT89S52 有计时计数器 Timer2，而 AT89S51 没有计时计数器 Timer2，所以对于 AT89S51 而言，此位保留不用。

假设我们要使用外部中断 INT0，此时必须先设置好使用 INT0 中断，设置的方式如下：

- (1) 设置 EA=0 以暂停整体中断；
- (2) 设置 EX0=1，启动 INT0 中断；
- (3) 设置 EA=1 以启动整体中断。

其中 EA 设置整体中断的启动或关闭，所以 EA=0 设置系统不接受所有中断，而 EA=1 设置系统可以接受中断。

EX0 设置接受或不接受外部中断 INT0，所以 EX0=0 设置不接受外部中断 INT0，而 EX0=1 设置接受外部中断 INT0。

同理,如果我们要使用外部中断 INT1 时,就必须把 EX0 改成 EX1,因为 EX1 设置接受或不接受外部中断 INT1,所以 EX1=0 设置不接受外部中断 INT1,而 EX1=1 设置接受外部中断 INT1。

3. 中断优先次序寄存器 IP 的内容

中断优先次序寄存器 IP 用来设置 AT89S5X 接受各种中断的优先顺序,当 AT89S5 同时接收到各项中断时,会根据此寄存器的内容判断应该先接受那一个中断。如图 1-15 所示先列出中断优先次序寄存器 IP 的内容。

位	7	6	5	4	3	2	1	0
	—	—	PT2	PS	PT1	PX1	PT0	PX0

图 1-15

假设我们同时使用外部中断 INT0 和外部中断 INT1,但是 INT0 引脚接到比较重要的信号源,则此时可以设置 PX0=1,也就是设置外部中断 INT0 优先被接受。

4. 计时器计数寄存器 TL0 和 TH0 的内容

AT89S5X 的内部包含两个 16 位的计时计数器,分别是 Timer0 和 Timer1。计时器计数寄存器 TL0 和 TH0 是储存计时计数器 Timer0 的当前计数内容,TL0 是低阶的 8 位,TH0 则是高阶的 8 位。TL0 和 TH0 组合起来就构成 16 位的计数器。每一个机械周期就会有一个时钟脉冲送到计时计数器 Timer0,此时计时计数器 Timer0 就往上加 1。当计时计数器 Timer0 在 16 位的模式下工作时,计时计数器 Timer0 计数到 65535 时,接下来的下一个脉冲就会让计时计数器 Timer0 产生溢出中断 (Overflow Interrupt),计时计数器 Timer0 内的数值会回到 0,如图 1-16 所示。

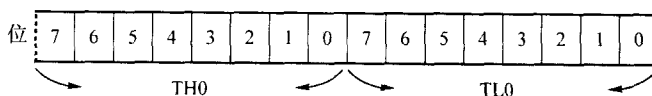


图 1-16

用户可以设置计时计数器 Timer0 在开始计数之前的内容。例如,如果 AT89S51 外接一颗振荡频率是 12MHz 的石英晶体时,因为 8051 的一个机械周期需要 12 个石英晶体的震荡周期,所以每秒就有 1 000 000 次的机械周期,换言之,1μs 就会有一个时钟脉冲送到计时计数器 Timer0。

如果我们希望 Timer0 每秒中断 200 次,那么我们就必须让 Timer0 每数 5000 次就中断 1 次 (1 000 000/200=5000)。因为 Timer0 的溢出中断是 Timer0 数到 65 536 (16 进制表示时是 10 000H) 就产生中断,因此要让 Timer0 数 5000 次就中断 1 次时就必须设置 Timer0 等于 10 000H - ((12 000 000/(12×200))),也就是 0xEE11。接下来我们就可以分别设置 Timer0 计数器的 TL0 和 Timer0 的 TH0,如下所示。

```
TL0=(0xEE11 & 0x00FF);    //设置 Timer0 计数器的低 8 位
TH0=(0xEE11 >> 8);        //设置 Timer0 计数器的高 8 位
```