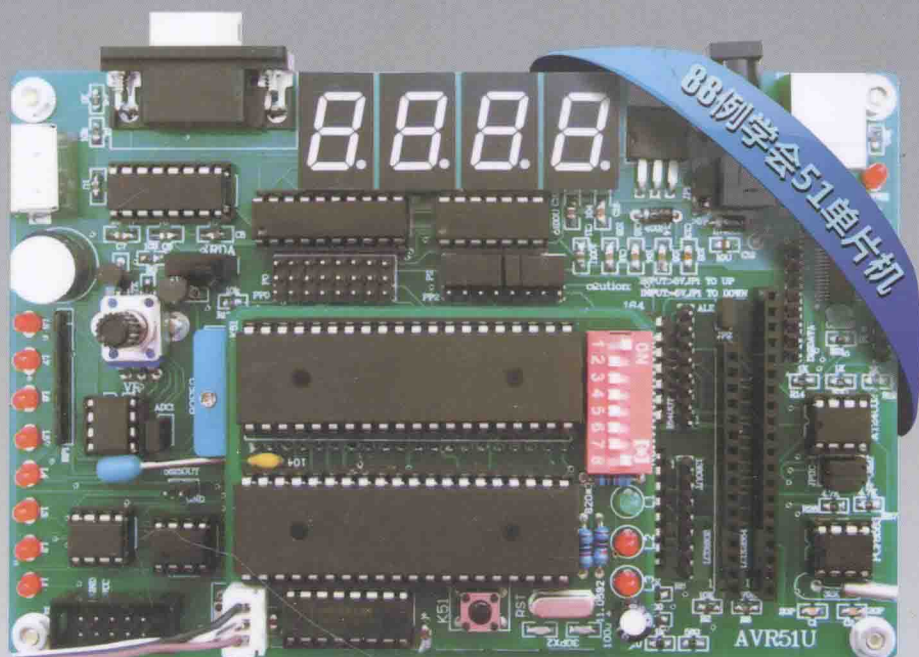




88例学会

51单片机

王建 宋永昌 主编



中国电力出版社
CHINA ELECTRIC POWER PRESS

88例学会 51单片机

主 编 王 建 宋永昌

副主编 王春晖 尚晓新 杨 峥 汪东震

参 编 张 波 季海峰 郝鑫虎 刘日晨

魏志强 费光彦 赵国利 孙锡保

梁少志 鄢 翔



内 容 提 要

本书以 51 系列单片机为核心,精选了 88 个设计开发实例,介绍了日常生活中典型和常见的控制系统的自主设计和实施,主要包括单片机对 LED 的控制、单片机的键盘控制、单片机的串行通信控制、单片机的显示控制、单片机的声音控制、单片机的外围器件控制。本书在选取实例时,对整体内容的深度和广度进行了梳理,开阔设计者的思路和视野。通过对本书的学习,读者可以顺利地完成任务、多功能单片机系统的设计和开发。

本书可作为从事单片机技术设计人员的参考用书,也可作为单片机技术提高的培训用书,还可供广大单片机爱好者、电气工作人员自学,以及高校相关专业师生参考。

图书在版编目 (CIP) 数据

88 例学会 51 单片机 / 王建, 宋永昌主编. —北京: 中国电力出版社, 2014. 5

ISBN 978 - 7 - 5123 - 5511 - 8

I . ①8… II . ①王… ②宋… III . ①单片微型计算机 - 基本知识 IV . ①TP368. 1

中国版本图书馆 CIP 数据核字 (2014) 第 024434 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2014 年 5 月第一版 2014 年 5 月北京第一次印刷

787 毫米 × 1092 毫米 16 开本 13 印张 312 千字

印数 0001—3000 册 定价 39.00 元



敬告读者

本书封底贴有防伪标签,刮开涂层可查询真伪

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

单片机具有体积小、价格低、控制灵活等特点，在智能仪器仪表、实时工业控制、交通及家用电器等领域得到了广泛的应用，从事单片机开发的人员也越来越多。在各本科院校，电气、信息类专业甚至机械类专业都开设了有关单片机的课程，全国大学生电子竞赛更是增加了单片机的项目。

当前单片机开发已由汇编语言编程逐步转为开发效率更高、程序可移植性更好的 C51 编程，片外存储芯片已由原来的并行接口芯片逐渐转换为串行接口芯片，各种新的开发技术不断涌现。本书内容的组织顺应单片机的发展趋势，兼顾入门读者和已经能进行开发的读者的需要，以 51 系列单片机为核心，精心选取了 88 个设计开发实例，对整体内容的深度和广度进行了梳理，列举了大量源程序，对编程技巧进行了讲解，程序的可移植性强。

本书以电气自动化实用技术为切入点，深入浅出地介绍了单片机对 LED、键盘、显示、声音、外围器件的控制，以及单片机的串行通信等，内容易学易理解，能够开阔设计者的思路和视野。通过对本书的学习，读者可以顺利地完成任务、多功能单片机系统的设计和开发。

本书的编写参考和借鉴了有关专家的论著、技术观点和许多宝贵的资料，在此对他们致以最衷心的感谢！

由于编者水平有限，书中的错误和不当之处，敬请读者批评指正。

编 者

2014 年 1 月

目 录

前言

第1章 单片机对 LED 的控制	1
例1 用单片机点亮第一盏灯	17
例2 闪烁灯	18
例3 不同闪烁时间的 LED	19
例4 右移的流水灯	20
例5 左移的流水灯	20
例6 左右循环流水灯	21
例7 自加方式控制 LED 流水花样	22
例8 地址方式控制 LED 流水灯	23
例9 利用按键控制 LED 的点亮	23
例10 单键控制 LED 的点亮状态	24
例11 循环语句控制 LED	26
例12 用 do-while 语句控制流水灯	26
例13 数组控制 LED 流水点亮	27
例14 指针形式的 LED 流水点亮 1	28
例15 指针形式的 LED 流水点亮 2	28
例16 指针形式的 LED 流水点亮 3	29
例17 函数参数控制 LED 流水速度 1	30
例18 函数参数控制 LED 流水速度 2	31
例19 函数参数控制 LED 流水速度 3	31
例20 T0 控制 LED 闪烁 1	32
例21 T0 控制 LED 闪烁 2	33
例22 T1 控制两个 LED 周期闪烁	33
例23 串行控制流水灯循环点亮	34
例24 花样流水灯	36
例25 单片机 PWM 控制 LED	38

例 26	LED 数码显示单一数字	42
例 27	LED 数码显示器循环显示数字	42
例 28	LED 数码管逐一显示数字	43
例 29	LED 数码动态显示数字	44
例 30	00 ~ 99 计数器	45
例 31	秒计时器	46
例 32	LED 模拟计数器	47
例 33	LED 八段码显示转速	48
例 34	74LS47 静态驱动 LED 显示	50
例 35	LED 模拟马表	51
第 2 章 单片机的键盘控制		53
例 36	独立式键盘控制形式 1	54
例 37	独立式键盘控制形式 2	54
例 38	独立式键盘控制形式 3	55
例 39	独立式键盘控制形式 4	57
例 40	独立式键盘控制形式 5	60
例 41	独立式键盘控制形式 6	62
例 42	独立式键盘控制数码时钟	63
例 43	独立式键盘控制步进电动机	67
例 44	矩阵式键盘控制形式 1	70
例 45	矩阵式键盘控制形式 2	72
例 46	按键控制数码管显示	73
例 47	按键控制数码管加减	75
例 48	矩阵键盘控制 LED 显示	76
例 49	数码管显示开关编码	78
例 50	独立式键盘控制继电器	79
例 51	PS2 键盘的读取	81
第 3 章 单片机的串行通信控制		88
例 52	单片机间的通信	88
例 53	数据发送形式 1	90
例 54	数据发送形式 2	91
例 55	数据接收形式 1	92
例 56	数据接收形式 2	93
例 57	单片机向 PC 发送数据	93
例 58	PC 向单片机发送数据	96

第4章 单片机的显示控制	98
例 59 8×8 LED 显示	98
例 60 点阵 LED 显示“0~9”	101
例 61 点阵式 LED 显示图形	102
例 62 用 LCD 显示字符	104
例 63 用 LCD 循环右移显示“Welcome to MCU”	111
例 64 用 LCD 显示模拟检测结果	113
例 65 用 12864 显示汉字	116
第5章 单片机的声音控制	123
例 66 单片机模拟报警声	123
例 67 “叮咚”门铃	125
例 68 单片机播放音乐	127
例 69 演奏音阶	128
例 70 按键控制选播音乐	129
第6章 单片机的外围器件控制	133
例 71 I ² C 总线 AT24C02 的操作	133
例 72 74LS138 译码器应用	139
例 73 74HC595 串入并出芯片应用	141
例 74 74LS165 实现串口扩展输入	144
例 75 74LS164 串口扩展输出	145
例 76 X5045 的操作	147
例 77 ADC0809 A/D 转换器基本应用技术	153
例 78 DAC0832 转换应用	158
第7章 单片机的综合应用	160
例 79 基于单片机的交通灯	160
例 80 基于单片机的 LED 时钟	162
例 81 基于单片机的简易电子琴	164
例 82 基于单片机的密码锁	166
例 83 基于单片机的无线遥控开关	171
例 84 基于单片机的红外遥控开关	176
例 85 基于单片机的液晶时钟	178
例 86 基于单片机的抢答器	183
例 87 基于单片机的智能充电器	186
例 88 基于单片机的数字温度计	191
参考文献	199

第 1 章

单片机对 LED 的控制

MCS-51 系列单片机研制于 1980 年, 由 Intel 公司所开发, 其结构是 8048 的延伸, 改进了 8048 的缺点, 其 ROM、RAM 都可扩充至 64KB, 也增加了如乘 (MUL)、除 (DIV)、减 (SUBB)、比较 (CJNE)、入栈 (PUSH)、出栈 (POP)、16 位数据指针、布尔代数运算等指令, 以及串行通信能力和 5 个中断源。8052 有 6 个中断源, MCS-51 系列单片机存储器容量见表 1-1。MCS-51 系列单片机特点如下:

- 专为控制应用所设计的 8 位 CPU;
- 具有布尔代码的运算能力;
- 32 条双向且可被独立寻址的 I/O 口;
- 芯片内有 128 字节可供存储数据的 RAM (8052: 256 字节);
- 内部有两组 16 位定时器 (8052 有 3 个);
- 具有全多工传输信号 UART;
- 5 个中断源, 且具有两级 (高/低) 优先权顺序的中断结构;
- 芯片内有 4KB (8KB/8052) 的程序存储器 (ROM);
- 芯片内有时钟 (CLOCK) 振荡器电路;
- 程序存储器可扩展至 64KB (ROM);
- 数据存储器可扩展至 64KB (RAM);
- 8051/52: 工厂烧写型, 内含 ROM;
- P8751: 一次烧写型, 内含 PROM;
- 8751/8752: 可重复烧写型, 内含 EPROM;
- 87C51/87C52: 省电型 (低消耗功率)。

表 1-1 MCS-51 系列单片机存储器容量 (字节)

型号	8031	8051	8751	8032	8052	8752
ROM	0	4K	0	0	8K	0
RAM	128	128	128	256	256	256
EPROM	0	0	4K	0	0	8K

1. 引脚图

引脚排列如图 1-1 所示。

2. 8051/8052 结构方框图

8051/8052 结构方框图如图 1-2 所示。

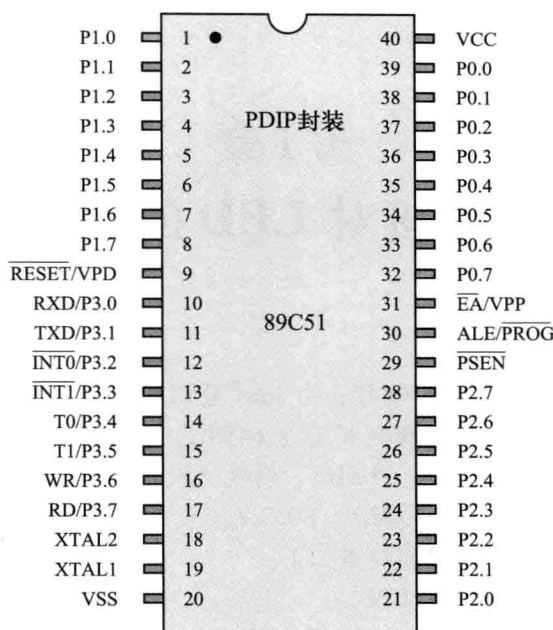


图 1-1 单片机外形图

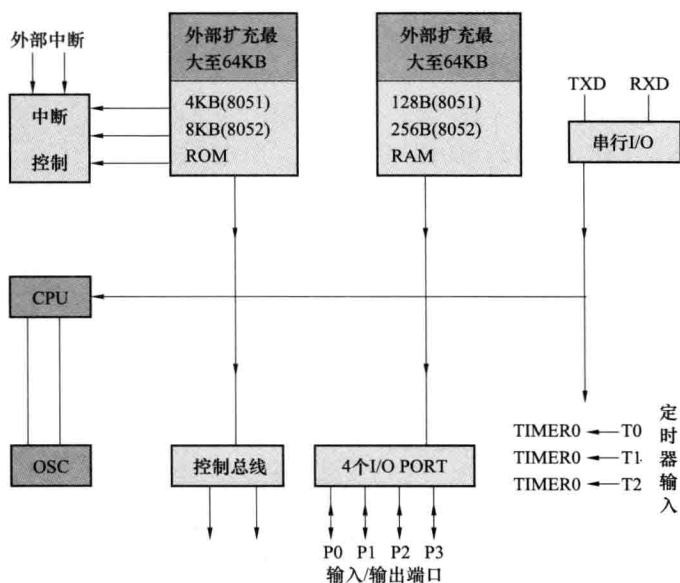


图 1-2 8051/8052 结构方框图

3. 8051 各引脚说明

I/O 端口：P0.0 ~ P0.7, P1.0 ~ P1.7, P2.0 ~ P2.7, P3.0 ~ P3.7。

8051 共有 4 个 I/O 端口，为 P0、P1、P2、P3，4 个 I/O 口都是双向的，且每个口都具有锁存器。每个口有 8 条线，共计 32 条 I/O 线。各端口的功能叙述如下。

(1) P0。P0 有 3 个功能：

1) 外部扩充存储器时, 当作数据总线 (D0 ~ D7)。
 2) 外部扩充存储器时, 当作地址总线 (A0 ~ A7)。
 3) 不扩充时, 可做一般 I/O 口使用, 但内部无上拉电阻, 作为输入或输出时应在外部接上拉电阻。

(2) P1。P1 只做 I/O 口使用, 其内部有上拉电阻。

(3) P2。P2 有 2 个功能:

1) 扩充外部存储器时, 当作地址总线 (A8 ~ A15) 使用。

2) 做一般 I/O 使用, 其内部有上拉电阻。

(4) P3。P3 有 2 个功能。除了作为 I/O 使用外 (内部有上拉电阻), 还有一些特殊的功能, 见表 1-2, 由特殊寄存器来设置。

表 1-2 端口引脚的特殊功能

端口的引脚	特 殊 功 能
P1.0 (8052)	T2 的外部输入引脚
P1.1 (8052)	$\overline{\text{T2EX}}$ (TIMER2) 的捕捉 (Capture) / 重新加载 (Reload) 的触发输出
P3.0	RXD (串行输入口)
P3.1	TXD (串行输出口)
P3.2	$\overline{\text{INT0}}$ (外部中断)
P3.3	$\overline{\text{INT1}}$ (外部中断)
P3.4	T0 (TIMER0 的外部输入脚)
P3.5	T1 (TIMER1 的外部输入脚)
P3.6	$\overline{\text{WR}}$ (外部数据存储器的写入控制信号)
P3.7	$\overline{\text{RD}}$ (外部数据存储器的读取控制信号)

端口 1、2、3 有内部上拉电路, 当作输入时, 其电位被拉高, 若输入为低电平可提供电流源, 其作为输出时可驱动 4 个 LS TTL。而端口 0 当作输入时, 处在高阻抗的状态。其输出缓冲器可驱动 8 个 LS TTL (需要外部的上拉电路)。

(5) VCC: 电源 +5V。

VSS: 接地。

(6) RESET: 此脚为高电平时 (约 2 个机器周期), 可将 CPU 复位, CPU 复位后其累加器和寄存器的内容见表 1-3。

表 1-3 CPU 复位后其累加器及寄存器的内容

寄存器	二进制值	寄存器	二进制值
ACC	00000000	IP	XXX00000
B	00000000	IE	0XX00000
PSW	00000000	TMOD	00000000
SP	00000111	TCON	00000000
P0/P1/P2/P3	11111111		

(7) $\overline{\text{ALE/PROG}}$ (ADDRESS LATCH ENABLE): 地址锁存使能信号端, 有 3 种功能:

1) 8051 外接 RAM/ROM: $\overline{\text{ALE}}$ 接地址锁存器 8252 (8212) 的 STB 脚, 74373 的 EN 脚, 当 CPU 对外部存储器进行存取时, 用以锁住地址的低位地址。

2) 8051 未外接 RAM/ROM: 在系统中未使用外部存储器时, $\overline{\text{ALE}}$ 脚也会有 1/6 石英晶体的振荡频率, 可作为外部时钟。

3) 在烧写 EPROM: $\overline{\text{ALE}}$ 作为烧写时钟的输入端。

(8) $\overline{\text{PSEN}}$ (PROGRAM STORE ENABLE): 程序储存使能端。

1) 内部程序存储器读取: 不动作。

2) 外部程序存储器读取 (ROM): 在每个机器周期活动做 2 次。

3) 外部数据存储器读取 (RAM): 两个 $\overline{\text{PSEN}}$ 脉冲被跳过不会输出。

4) 外接 ROM 时, 与 ROM 的 $\overline{\text{OE}}$ 脚连接。

(9) $\overline{\text{EA/VPP}}$ 。

1) 接高电平时:

① CPU 读取内部程序存储器 (ROM), 如 8051/8052。

② 扩充外部 ROM: 当读取内部程序存储器超过 0FFFH (8051)、1FFFH (8052) 时, 自动读取外部 ROM。

2) 接低电平时: CPU 读取外部程序存储器 (ROM), 如 8031/8032。

3) 8751 烧写内部 EPROM 时, 利用此脚输入 21V 的烧写电压。

(10) XTAL1、XTAL2: 接石英晶体振荡器。

机器周期 = 石英晶体周期 $\times 12$, 如 12MHz 石英晶体 / 12 = 1 μs 。

4. 程序存储器 ROM / EPROM

(1) MCS-51 程序存储器配置图。MCS-51 程序存储器配置图如图 1-3 所示。

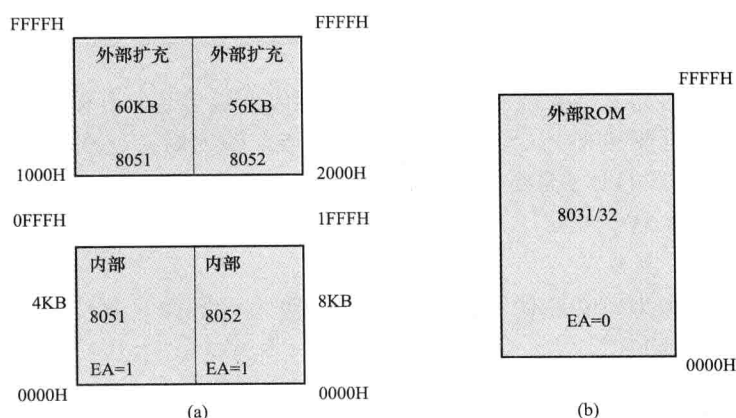


图 1-3 程序存储器配置图

(a) 8051/8052; (b) 8031/8032

(2) 程序存储器的 4KB (8052 8KB), 可以是芯片内部的 ROM 或外部的 ROM (EPROM), 其选择方式是由 MCS-51 的 $\overline{\text{EA}}$ 引脚电压来决定的。 $\overline{\text{EA}} = \text{VCC}$ 时是读取内部 ROM 的地址范围为 0000H ~ 0FFFH 4KB (8052 0000H ~ 1FFFH 8KB), 若超过 4KB (8052 8KB) 时, CPU 将自动到外部读取, 所以 8051/52 扩充 ROM 至 64KB。

(3) EA = 0 时, 完全读取外部 ROM, 可扩充至 64KB (如 8031、8032), 另外当读取外部程序存储器时, 会使能 $\overline{\text{PSEN}}$, 使其变为低电平, 以使外部程序存储器使能; 而读取内部程序时, $\overline{\text{PSEN}}$ 将保持高电平。通常 $\overline{\text{PSEN}}$ 与外部 ROM 的 $\overline{\text{OE}}$ 相连接。

(4) 读取外部程序存储器时, P0 口及 P2 口的 16 条 I/O 线作为地址总线, 其中 P0 是多工的, 它送出低 8 位的地址码, 也读取指令代码。工作时 P0 首先送出低 8 位地址码, 然后变成浮接 (高阻抗), 等待读取外部程序存储器送出的指令码, 所以当 P0 送出低 8 位地址码时, CPU 的地址锁存使能 “ALE” 也送出使能信号, 使地址锁存器 (8280/8212/74373) 使能, 而将低 8 位地址码锁住。然后在 P0 送出低 8 位地址码时, P2 也同时送出高 8 位地址码。由于 P2 并非多工, 所以 P2 不需要加地址锁存器。当 16 位地址寻址后, $\overline{\text{PSEN}}$ 送出一个 LO 信号使外部 ROM 使能, 外部 ROM 会将指令码送出 CPU 的指令译码器, 译码并执行该指令。

(5) 当 MCS-51 的 CPU 被复位后, 会从 0000H 处开始执行程序。每个中断在程序存储器中都有一个对应指令的地址, 当中断产生并接收时, 会使 CPU 跳至该地址开始执行中断子程序, 具体中断对应地址见表 1-4。

表 1-4 各中断对应的地址

中断名	起始地址	说 明
—	0000H	系统复位起始地址
$\overline{\text{INT0}}$	0003H	外部中断 $\overline{\text{INT0}}$ 矢量地址
TIMER0	000BH	定时/计数 TIMER0 中断矢量地址
$\overline{\text{INT1}}$	0013H	外部中断 $\overline{\text{INT1}}$ 矢量地址
TIMER1	001BH	定时/计数 TIMER1 中断矢量地址
UART	0023H	串行口中断矢量地址
TIMER2 (8052)	0002H	定时/计数 TIMER2 中断矢量地址

5. 数据存储器 RAM

MCS-51 系统中 8031/51 内部数据存储器有 128 字节, 8032/52 有 256 字节, 外部扩充 RAM 最大可至 64KB。

(1) 内部数据存储器分成三部分, 如图 1-4 所示。

- 较低地址 128 字节 (00 ~ 7FH) 的数据存储器, 如图 1-5 所示。
- 较高 128 字节 (80 ~ FFH) 的数据存储器, 如图 1-6 所示。
- 特殊功能寄存器 (Special Function Register, 简称 SFR)。

1) MCS-51 的较低的 128 字节, 其中最低的 32 字节 (00 ~ 1FH), 被分成 4 个寄存器组 (Register Bank), 分别为 RB0、RB1、RB2、RB3, 由程序状态字寄存器 (PSW) 中的 RS1、RS0 来选择对哪一个寄存器组进行操作。这 128 字节的 RAM, 其存取可用直接寻址方式, 如:

MOV 30H, A ; 将累加器的值加载到地址为 30H 的寄存器中

MOV A, 30H ; 将地址为 30H 寄存器中的值加载到累加器中

间接寻址如:

MOV R0, #30H ; 寻址在 30H 地址的寄存器

MOV @R0, A ; 将累加器的值加载到 R0 所指定 RAM 的地址

MOV R0, #30H ; 寻址在 30H 地址的寄存器

MOV A, @R0 ; 将 R0 所指定 RAM 地址的内容加载到累加器中

且 RAM 20H ~ 2FH 又可直接位寻址, 如 SETB 20H. 0、CLR 20H. 1。



图 1-4 内部数据存储器分布空间

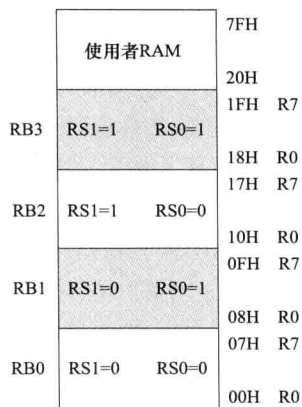


图 1-5 4 个寄存器组的地址范围

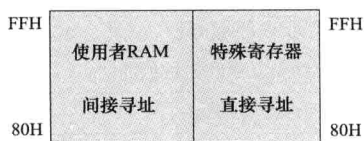


图 1-6 8032/52 中的高 128 字节

2) 高 128 字节: 8031/51 中不存在, 8032/52 在存取时只能用间接寻址的方式存取数据, 如: MOV @R0, A (存入), MOV A, @R0 (取出), 而无法以直接寻址方式存取。其所在地址 80H ~ FFH 与特殊功能寄存器似乎相同, 但实际是两个不同的存储空间, 间接寻址存取 RAM, 而直接寻址是特殊功能寄存器。

3) 特殊功能寄存器 (SFR) 的地址, SFR 中包含 I/O 口的锁存器 (LATCH)、定时器和外围控制寄存器等, 而这些寄存器只能以直接寻址的方式进行存取, 如图 1-7 所示。

(2) 特殊功能寄存器说明如下。

1) B 寄存器: 乘除运算时, 存放乘积的高位字节或除法的余数, 也用作一般存储器。

2) 累加器 ACC: 用于一般数据转移或运算结果的存放。

3) 程序状态字 (PSW): 用来存放 CPU 状态, 如图 1-8 所示, 改变它的值可以改变 CPU 的状态。

① RS0、RS1 工作寄存器组选择位。

② 进位标志位 CY, 作用有两种:

- 算术运算中进位或借位时, 设定 CY = 1。
- 作为布尔代数位处理的累加器。

③ 溢出标志位 OV, 当进行正负号数值的加减运算时, 如结果超出 7 位或者 2 的补码 (+127 ~ -128) 时, OV = 1。

④ 奇偶校验标志位 P, 当累加器中有奇数个 1, 则 P = 1; 偶数个 1, 则 P = 0。在串行传输时用来检测传送是否有错。

4) IP 中断优先权控制寄存器, 如图 1-9 所示。

PX0: 定义外部中断 INT0 的优先权。

PT0: 定义 TIMER0 的优先权。

FFH									
F0H	F7	F6	F5	F4	F3	F2	F1	F0	B寄存器
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC累加器
D0H	CY	AC	F0	RS1	RS0	OV	P		
	D7	D6	D5	D4	D3	D2	D1	D0	PSW状态字
B8H				PS	PT1	PX1	PT0	PX0	
	-	-	-	BC	BB	BA	B9	B8	IP中断有限权控制寄存器
B0H									
	B7	B6	B5	B4	B3	B2	B1	B0	P3口3寄存器
A8H				ES	ET1	EX1	ET0	EX0	
	AF	-	-	AC	AB	AA	A9	A8	IE中断使能控制寄存器
A0H									
	A7	A6	A5	A4	A3	A2	A1	A0	P2口2寄存器
98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
	9F	9E	9D	9C	9B	9A	99	98	SCON串行控制寄存器
90H									
	97	96	95	94	93	92	91	90	P1口1寄存器
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
	8F	8E	8D	8C	8B	8A	89	88	TCON定时器/计数器控制寄存器
80H									
	87	86	85	84	83	82	81	80	P0口0寄存器

(a)

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

(b)

图 1-7 特殊功能寄存器 (SFR) 的存储器分布

(a) SFR 的地址及其位对应地址的分布; (b) SFR 的地址分布

CY	AC	F0	RS1	RS0	OV	-	P
进位标志位	辅助进位标志位	一般用途	寄存器组选择位	寄存器组选择位	溢出标志位	保留	奇偶校验标志位
PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0

图 1-8 程序状态字各位的含义

PX1: 定义外部中断 $\overline{\text{INT1}}$ 的优先权。

PT1: 定义 TIMER1 的优先权。

PS: 定义串行口的优先权。

PT2: 定义 TIMER2 的优先权。

说明: 1 表示设定为允许, MOV IP, #00000001B 表示 INT0 为中断优先。

—	—	PT2	PS	PT1	PX1	PT0	PX0	IP 中断优先权控制寄存器
---	---	-----	----	-----	-----	-----	-----	---------------

图 1-9 IP 中断优先权控制寄存器

5) IE 中断使能控制寄存器, 如图 1-10 所示。

EA	—	ET2	ES	ET1	EX1	ET0	EX0	IE 中断使能控制寄存器
----	---	-----	----	-----	-----	-----	-----	--------------

图 1-10 IE 中断使能控制寄存器

EX0: 使能外部 $\overline{\text{INT0}}$ 中断。

ET0: 使能 TIMER0 中断。

EX1: 使能外部 $\overline{\text{INT1}}$ 中断。

ET1: 使能 TIMER1 中断。

ES: 使能串行口中断。

ET2: 使能 TIMER2 溢出或压入的中断 (8052)。

EA: 如果 EA = 0 时, 屏蔽所有的中断; 如果 EA = 1, 则各中断由各个中断位加以设定。

说明: 1 表示设定为使能, MOV IE, #10010001B 表示设定 INT0 外部中断、串行口中断。

6) SCON 串行控制寄存器, 如图 1-11 所示。

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

图 1-11 SCON 串行控制寄存器

RI: 接收中断标志位, 在模式 0 下, 当第 8 位结束时, 硬件会将其设为 1; 在其他模式时, 在停止位的一半时由硬件设定 (参考 SM2), 此位必须由软件清除。

TI: 传送中断标志位, 在模式 0 下, 当第 8 位结束时, 硬件会将其设为 1; 在其他模式时, 在停止位的开始时由硬件设定 (参考 SM2), 此位必须由软件清除。

RB8: 在模式 2 或 3 时, 传送的第 9 位数据位放入此位; 在模式 1 时, 若 SM = 0 时, 则 RB8 为接收到的停止位。模式 0 时, RB8 没有作用。

TB8: 在模式 2 或 3 时, 传送时的第 9 数据位, 由软件控制。

REN: 由软件设定或清除, 以决定是否接受串行输入数据 (REN = 1 接收)。

SM2: 在串行端口模式 2 或 3 时, 使能多处理器通信的功能; 在模式 2 或 3 时, 如果 SM2 = 1, 则当接收到第 9 数据位为 0 时, RI 不动作; 在模式 1 时, 如果 SM2 = 1, 当接收到的停止位不正确时, RI 也不动作; 在模式 0 时, SM2 必须为 0。

SM1: 串行端口模式选择。

SM0：串行端口模式选择。

7) TMOD 定时器模式控制寄存器（地址：89H），如图 1-12 所示。

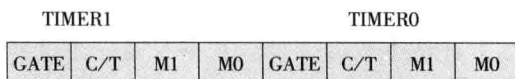


图 1-12 TMOD 定时器模式控制寄存器

GATE：当 TRX（在 TCON 内）=1 且 GATE=1 时，则定时器只在 $\overline{\text{IN1X}}$ 引脚为高电平时才会计时；当 GATE=0 时，则定时器只在 TRX=1 时才会计时。

C/T：C/T=0 时为定时器，C/T=1 时为计数器。

M1、M0：工作模式选择。

8) PCON：电源控制寄存器（POWER CONTROL REGISTER），如图 1-13 所示。



图 1-13 电源控制寄存器 PCON

SMOD：双倍波特率标志位，当串行工作于模式 1、2 或 3 时，如使用 TIMER1 作为波特率发生器，且 SMOD=1，则波特率为双倍。

GF1：一般用途。

GF0：一般用途。

PD：待机位。设定此位单片机进入能耗最低的待机模式（仅 CHMOS 可以）。PD 和 IDL 同时为 1 时，PD 较优先。

6. 数据类型

任何程序设计都离不开对数据的处理，一个程序如果没有数据，就没有办法工作。

数据在计算机内存中的存放情况是由数据类型决定的。在 C 语言中分基本数据类型和复杂数据类型，复杂数据类型是由基本数据类型构造而来的，比如结构。因为单片机程序中较少使用，所以只介绍基本数据类型。

C 语言基本数据类型有：char、int、short、long、float 和 double。对于 C51 编译器来说 short 和 int 类型相同，long 和 double 类型相同，所以可以统一标识，这样只剩下：char、int、long 和 float，其中前 3 种还有有符号和无符号之分，表示数值的范围不同，默认情况下都是有符号数，无符号数的表示要在数据类型前加 unsigned。

(1) char：字符类型。

它表示的数据的长度是一个字节，也就是 8 位数。对于有符号数 char，表示数值范围是 -128 ~ +127。对于无符号数 unsigned char，表示数值范围是 0 ~ 255。

(2) int：整型。

它表示的数据的长度是 2 个字节，也就是 16 位数。对于有符号数 int，表示数值范围是 -32768 ~ +32767。对于无符号数 unsigned int，表示数值范围是 0 ~ 65535。

(3) long：长整型。

长整型表示的数据的长度是 4 个字节，在单片机中这种类型比较少使用，尤其是 8 位机中。

(4) float: 浮点型。

它是单精度浮点数据类型。一个 float 类型数据要占用 4 个内存字节。它没有符号之分。

(5) *: 指针型。

指针型数据不同于以上 4 种基本数据类型。它本身是一个变量,但在这个变量中存放的不是普通的数据,而是指向另一个数据的地址。它也要占用一定的内存单元。在 C51 中,指针的长度一般为 1~3 个字节。

指针变量也是具有类型的,它的类型表示该指针所指向的地址中数据的类型,比如:char * pt1 表示 pt1 是一个字符型指针变量,也就是 pt1 所指向的地址中的数据类型是字符型。

(6) bit: 位类型。

这是 C51 编译器的一种扩展类型。在 51 单片机的内存区域有一块 128 位的内存单元,其中每个位都可以定义成位变量,但不能定义为位指针或位数组。

(7) sfr: 8 位特殊功能寄存器类型。

这也是 C51 编译器的一种扩展类型。利用它可以定义单片机中所有 8 位特殊功能寄存器。

(8) sfr16: 16 位特殊功能寄存器类型。

利用它可以定义单片机中所有 16 位特殊功能寄存器。

(9) sbit: 可寻址位。

单片机中的特殊功能寄存器有许多是可以位寻址的,这些可位寻址的寄存器都有个特点,就是它的地址可以被 8 整除。比如 P0,它的地址是 0x80,就可以这样定义:

```
sfr    P0 = 0x80;           //将 P0 口地址定义为 80H
sbit   P02 = P0 ^ 2;        //将 P0.2 位定义为 P02
```

7. 运算符

C 语言有丰富的运算符,正是这些运算符,使得对数据的操作非常灵活,而编译成目标代码后,具有较高的代码效率。

(1) 赋值运算符: =。

把一个常量或变量或运算式的值传递给另一个变量。比如: tmp1 = 30; 就是把值 30 赋值给变量 tmp1。

(2) 关系运算符: ==、!=、>、<、>=、<=。

用来比较两个表达式之间的关系,比较的结果有两种: 0 = 不成立; 1 = 成立。

```
比如: if (tmp1 == tmp2)
    语句 1;
    else
    语句 2;
```

如果比较的结果成立,则执行语句 1,不成立则执行语句 2。

(3) 增量与减量运算符: ++、--。

使变量值自动加 1 或减 1,运算符放在变量的前后意义不同。