

李勋 林广艳 卢景山 编著

单片微型计算机

大学读本

68.1
-5

北京航空航天大学出版社

单片微型计算机

大学读本

李 勋 林广艳 卢景山 编著

北京航空航天大学出版社

JS/55/16

内 容 简 介

本书分基础、加强和特色三篇九章。它以基础篇为主,全面系统地讲解了 MCS-51 系列单片机的硬件组成、运作原理、指令集、软硬件应用技术以及系统设计等基本知识;加强篇则以高档 8 位单片机 8XC552 为代表机型,详细阐述了它的特点、新增硬件资源的组成原理及应用技巧;特色篇扼要地介绍了 EPROM 型单片机的编程方法、程序封锁措施以及闪电存储器型 AT89C51 系列单片机的特点与典型应用。书末附有习题库,供师生选用。

本书内容新颖、取材精练、资料翔实,是一部紧跟时代脚步的大学教材。它条理清晰、文笔流畅、逻辑性强、例题丰富、题库精深、可读性好,是大专院校有关专业师生及从事微机控制的广大科技人员的一本物美价廉的读物。

图书在版编目(CIP)数据

单片微型计算机大学读本/李勋等编著·一北京:北京航空航天大学出版社,1998.11

ISBN 7-81012-831-0

I. 单… II, 李… III. 单片微型计算机-高等教育-学习参考资料 IV. TP368.1

中国版本图书馆 CIP 数据核字(98)第 25701 号

单 片 微 型 计 算 机 大 学 读 本

李 勋 林广艳 卢景山 编著
责任编辑 杨昌竹
责任校对 李宝田

*

北京航空航天大学出版社出版发行

北京市学院路 37 号,邮编 100083 发行部电话 62015720

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

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

河北省涿州市新华印刷厂印装 各地书店经销

*

开本: 787×1092 1/16 印张: 13.5 字数: 350 千字

1998 年 11 月第 1 版 1998 年 11 月第 1 次印刷 印数: 5 000 册

ISBN 7-81012-831-0/TP·309 定价: 18.00 元

前 言

几年前,出版社杨昌竹教授曾高瞻远瞩地提出,为适应我国高等教育新的发展形势,应全面更新单片微型计算机读物的内容,编写一本包括该领域最新发展技术、紧跟时代步伐、适于课堂教学需要的大学教材。笔者当时未敢领命,但此后的数年间,却也一直做着踏踏实实的准备工作——深入钻研了以工业标准 8051 为基础而发展起来的高档 8 位单片机 8XC552 和 AT89C51 系列之有关的资料,继而以此两种器件为微控制器完成了几个科研项目。于是,在《日立 H8/3048 系列单片机应用技术》一书交稿之后,我们便着手进行本书的编写工作。所以,《单片微型计算机大学读本》的问世,首先应感谢杨昌竹教授的鼓励和支持。

本书实际上也是十年前我们在天津科技翻译出版公司出版的《MCS-51 单片微型计算机》一书的发展与继续。该书发行后,作者收到过全国各地读者的许多信函,普遍对之表示欢迎。至今我们仍然记得,山西一位老师在来信中对那本小书给了很高的评价。因此,《单片微型计算机大学读本》一书也是广大读者鼓舞和鞭策的产物。

全书共分基础、加强和特色三篇,计九章,由李勋教授(第一、五、八章及附录)、林广艳老师(第三、六、七章)、卢景山老师(第四、九章)和李克田老师(第二章)共同编写,最后由李勋教授统一修改定稿。

基础篇全面系统地讲解了 MCS-51 系列单片机,尤其突出了 80C51 子系列的组成原理,例示了软、硬件应用技术。这是读者掌握并进

而应用单片机的基础。加强篇详细阐述了当前功能最强的 8 位单片机系列 8XC552 的特点,深入透彻地研究了其新增硬件资源的运作原理及应用技巧。这是读者深化单片机知识、提高实践技能的首选内容。特色篇扼要地介绍了 EPROM 型单片机的编程方法、程序封锁措施以及闪电(闪速)存储器型微控制器的特点。AT89C51 闪电存储器型单片机是一种很有特色的器件,它的擦写要比 EPROM 型方便得多。

《单片微型计算机大学读本》可作大专院校有关课程的教材。对于自动化、计算机以及机电一体化等专业,最好基础、加强和特色三篇各有侧重地全面讲授;其他本科专业学生可以基础篇为主,扼要地掌握其后两篇之内容;专科生只学基础篇,其他两篇留作参考。从事微机控制的广大科技人员,若欲系统巩固、阶段更新其单片机知识,提高软、硬件设计技能,阅读此书定会大有收获。

加强篇和特色篇所涉及的内容,是在作者吃透原文精神,并经过数个项目的实践后写出的。读者可放心地阅读和引用,保证不会被误导,不会造成以讹传讹。全书各关键环节均配有大量例题,这大大提高了它的可读性、可操作性和实用性。同时,这些例题有助于打开读者的思路,及时化解阅读过程中所产生的疑难问题,加深对原理的进一步理解和掌握。书后所附题库乃作者多年教学工作的心血结晶,我们深信,它将是老师命题和学员知识自检的珍贵资料。

300160 天津纺织工学院自动化系

李 勋 1998.4

目 录

基 础 篇

第一章 MCS-51 硬件结构..... 1

1.1 引言.....	1
1.2 MCS-51 系列.....	1
1.3 内部结构.....	2
1.3.1 专用寄存器.....	2
1.3.2 端口结构及运作.....	5
1.3.3 存储器组织.....	9
1.3.4 CPU 定时.....	11
1.3.5 片外存储器的存取.....	12
1.3.6 复位电路.....	14
1.3.7 片内时钟振荡器.....	16
1.4 定时/计数器.....	17
1.4.1 定时器 0 和定时器 1.....	17
1.4.2 定时器 2.....	19
1.5 串行口.....	21
1.5.1 运作方式.....	21
1.5.2 串行口控制寄存器.....	22
1.5.3 多机通讯.....	23
1.5.4 波特率.....	23
1.6 中断系统.....	25
1.6.1 中断源.....	25
1.6.2 中断允许和优先级寄存器.....	26
1.6.3 中断处理过程.....	28
1.6.4 外部中断的触发方式.....	29
1.6.5 中断响应时间.....	30
1.7 引脚功能.....	30

第二章 指令系统..... 33

2.1 寻址方式.....	33
2.1.1 寄存器寻址.....	33
2.1.2 直接寻址.....	33
2.1.3 寄存器间接寻址.....	33
2.1.4 立即寻址.....	33
2.1.5 基址寄存器加变址寄存器间接寻址.....	34
2.2 指令分类.....	34
2.3 数据传送指令.....	34
2.3.1 单向数据传送指令.....	34
2.3.2 数据交换指令.....	38
2.4 算术指令.....	39
2.4.1 加法类指令.....	39
2.4.2 减法类指令.....	41
2.4.3 乘除法指令.....	42
2.5 逻辑指令.....	42

2.5.1 单操作数逻辑运算指令.....	42
2.5.2 双操作数逻辑运算指令.....	44
2.6 控制跳转指令.....	46
2.6.1 子程序调用和返回指令.....	46
2.6.2 无条件跳转指令.....	46
2.6.3 条件跳转指令.....	48
2.6.4 比较不等则跳转指令.....	49
2.6.5 循环控制及其他指令.....	49

第三章 程序设计示范..... 51

3.1 数制转换子程序.....	51
3.2 多倍精度运算.....	51
3.3 查表程序.....	52
3.4 堆栈操作程序.....	53
3.4.1 堆栈在中断时的应用.....	53
3.4.2 堆栈在参数传递中的应用.....	55
3.5 分支程序.....	56
3.5.1 目的地址的查表法.....	56
3.5.2 目的地址的实时计算法.....	57
3.6 代码排队参数传递法.....	58

第四章 外围接口技术..... 60

4.1 接口软件实例.....	60
4.1.1 虚拟 I/O 端口.....	60
4.1.2 控制信号的软件定时.....	62
4.1.3 串行口和定时器的方式设定.....	62
4.1.4 简单的串行 I/O 驱动程序.....	62
4.1.5 串行口字符串的传送.....	63
4.1.6 特别情况的辨识与处理.....	63
4.1.7 多机串行通讯.....	64
4.1.8 定时器中断示例.....	66
4.1.9 定时溢出和中断的同步.....	66
4.1.10 定时器瞬时值的动态读出.....	67
4.1.11 单步运行.....	67
4.1.12 中断优先级别的扩展.....	68
4.2 外围接口实例.....	68
4.2.1 程序存储器的扩展.....	68
4.2.2 数据存储器的扩展.....	70
4.2.3 程序/数据存储器的扩展.....	71
4.2.4 片外数据存储器和 I/O 扩展.....	73
4.2.5 多中断源的安排.....	79
4.2.6 8031 最小用户系统.....	80
4.2.7 I/O 口扩展.....	80
4.2.8 并行 I/O 口的直接应用.....	86

第五章 布尔处理功能应用实例..... 88

5.1 布尔变量逻辑函数的软件解法.....	88	7.3.2 引脚功能.....	134
5.2 汽车转弯信号灯的控制.....	88	7.3.3 器件型号.....	136
5.3 复杂控制功能的实现.....	92	第八章 8XC552 片内硬件资源.....	137
5.3.1 输入矩阵的实施方案.....	93	8.1 定时器 T2.....	137
5.3.2 组合输出变量的计算.....	95	8.1.1 内部结构及运作原理.....	137
5.3.3 中间变量的计算.....	96	8.1.2 定时器 2 控制寄存器 TM2CON.....	138
5.3.4 和远方处理器的通讯及其他.....	97	8.1.3 输入捕捉逻辑.....	140
第六章 80C51BH 和 80C52.....	99	8.1.4 输出比较逻辑.....	141
6.1 引言.....	99	8.1.5 中断标志寄存器 TM2IR.....	143
6.1.1 CMOS 的演化.....	99	8.1.6 中断优先级寄存器 IP1.....	144
6.1.2 CHMOS 工艺.....	99	8.1.7 中断允许寄存器 IEN1.....	144
6.1.3 MCS-51 系列中的 CHMOS 品种.....	100	8.1.8 定时器 T2 的应用.....	144
6.2 CMOS 和 CHMOS 电路的特点.....	100	8.2 定时器 3——监视定时器.....	146
6.2.1 门连现象.....	100	8.2.1 内部结构及运作原理.....	146
6.2.2 逻辑电平与接口问题.....	100	8.2.2 监视定时器的使用细则.....	147
6.2.3 抗干扰能力.....	101	8.2.3 监视定时器软件示例.....	148
6.2.4 未用引脚的处理.....	102	8.3 串行 I/O 口.....	148
6.2.5 上拉电阻.....	102	8.3.1 I ² C 串行 I/O 口概要.....	148
6.2.6 下拉电阻.....	103	8.3.2 运作方式.....	149
6.2.7 内部上拉器件的驱动能力.....	104	8.3.3 SIO1 的内部结构.....	150
6.2.8 功耗.....	104	8.3.4 SIO1 的专用寄存器.....	152
6.3 80C51BH 应用技巧.....	107	8.3.5 运作方式图解.....	156
6.3.1 掉电方式的应用.....	107	8.3.6 SIO1 服务程序示例.....	165
6.3.2 电池后备系统.....	108	8.4 I/O 端口结构.....	171
6.3.3 电源切换电路.....	110	8.5 脉冲宽度调制输出.....	173
6.3.4 80C31BH 配用 CHMOS EPROM.....	110	8.6 A/D 转换器.....	174
6.3.5 键盘扫描.....	112	8.6.1 A/D 转换.....	174
6.3.6 驱动液晶显示器.....	114	8.6.2 ADC 分辨率及模拟电压.....	176
6.3.7 LCD 驱动器.....	115	8.7 中断系统.....	177
6.3.8 频率测量.....	118	8.7.1 中断源.....	177
6.3.9 周期测量.....	119	8.7.2 中断允许寄存器.....	178
6.3.10 脉冲宽度测量.....	120	8.7.3 中断优先级寄存器.....	179
6.3.11 HMOS/CHMOS 的互换性.....	121	8.7.4 中断向量.....	180
6.4 80C52.....	123	8.8 低功耗运作方式.....	180
6.4.1 定时器 2.....	123	特 色 篇	
6.4.2 串行口.....	124	第九章 AT89C51 及其他.....	183
6.4.3 串行口应用示例.....	125	9.1 EPROM 型器件.....	183
6.4.4 掉电方式.....	126	9.1.1 8751H.....	183
6.4.5 断电标志.....	127	9.1.2 8751BH/8752BH 和 87C552.....	185
6.4.6 在线仿真方式.....	127	9.1.3 87C51.....	188
加 强 篇		9.2 闪电存储器型器件.....	189
第七章 8XC552 概要.....	129	9.2.1 89C51.....	190
7.1 概述.....	129	9.2.2 AT89C2051.....	193
7.2 内部结构.....	130	9.3 串行 EEPROM 和 AT89C2051 接口.....	197
7.2.1 存储器组织.....	130	9.3.1 硬件连接.....	197
7.2.2 专用寄存器.....	130	9.3.2 双向数据传送协议.....	198
7.2.3 片内振荡器和复位电路.....	131	9.3.3 模拟 I ² C 总线数据传送.....	199
7.3 引脚配置及封装.....	133	附录 题库.....	204
7.3.1 引脚排列.....	133	参考文献.....	210

基 础 篇

第一章 MCS-51 硬件结构

1.1 引 言

单片微型计算机是把 CPU、一定容量的存储器以及若干输入 / 输出接口等部件集成在一块小小硅片上的微型计算机, 简称单片机。

自从 1976 年美国 Intel 公司推出了第一代 8 位通用 MCS-48 系列单片机以来的 20 多年间, 单片机技术蓬勃发展, 质量和性能逐年提高, 应用领域日益扩大。目前世界上各大半导体公司所开发并生产的单片机系列, 各具特色, 阵容强大, 品种齐全。据信有几十个系列, 数百余品种, 可满足各类系统设计的需求。单片机一问世, 就以其体积小、功能全和价格低等优点赢得了市场的欢迎和用户的青睐。目前它们正在过程控制、智能仪表、机电一体化和家用电器等领域中发挥着重要的控制作用。因此, 单片机也称微控制器。

微控制器虽然品种繁多, 但就我国乃至世界应用范围来看, 功能最强者, 16 位机当属日立公司的 H8/3048 系列, 8 位机要算是 Intel 公司的 MCS-51 系列了, 尤其是 Philips 公司的 80C51 及其派生产品, 在我国受到了普遍欢迎。ATMEL 公司的闪电存储器型微控制器 AT89C51 更是后来居上, 大有取代 MCS-51 系列中 EPROM 型产品之势。

1.2 MCS-51 系列

早期的 MCS-51 微控制器系列包括下列各品种。

8051 这是 MCS-51 家族的最初成员, 1981 年开始生产。其特点如下:

- 最适于控制用的 8 位 CPU;
- 强化的布尔(单位逻辑)处理能力;
- 32 条可按位寻址的双向 I/O 线;
- 128 个字节的片内数据存储器 RAM;
- 两个 16 位的递增定时 / 计数器;
- 全双工的通用异步接收 / 发送器(UART);
- 两个优先级别的五源中断结构;
- 片内时钟振荡器;
- 4 k 字节片内程序存储器;
- 64 k 字节程序存储器地址空间;

- 64 k 字节数据存储器地址空间。

8051 尚有另外两个变体，即无片内程序存储器的 8031 和有片内 EPROM 的 8751。但后者已不再生产，而被 8751H 所取代。

8051AH 此芯片系以当时较新的 HMOSII 技术制造而成，其他方面与 8051 完全相同。8031AH 为无片内 ROM 的 8051AH；8751AH 则是以 EPROM 取代了掩膜 ROM 的 8051AH。

8052AH 8052AH 为增强型的 8051，也是采用 HMOSII 技术制造的，与 8051 向上兼容。其胜过 8051 之处在于：

- 256 个字节的片内 RAM；
- 三个定时 / 计数器；
- 六源中断结构；
- 8 k 字节的片内程序存储器。

8752BH 和 8032AH 分别为带片内 EPROM 和无片内 ROM 之变体品种。8052 尚有一个单独的品种，叫作 8052AH-BASIC，其片内 ROM 中驻留有全 BASIC 解释程序。

80C51BH 8051 的 CHMOS 变体称作 80C51BH，两者功能完全兼容。但是大家知道，CMOS 电路之特点在于其功耗低，既然 80C51BH 是以 CHMOS，即高速 CMOS 生产的，那么，它当然就比与其同功能的 HMOS 芯片 8051 耗电要少。

80C51BH 之片内 EPROM 和无片内 ROM 的两种变体品种分别为 87C51 和 80C31BH。

表 1-1 中简要地汇总了上述各器件的情况。对于这些芯片，若不特指某一具体型号，为方便起见，可只称 8051 和 8052。前者包括 8051、8051AH、80C51BH 及与它们相对应的无片内 ROM 和带片内 EPROM 型之诸品种；8052 则泛指 8052AH、8032AH 和 8752BH 等。

表 1-1 MCS-51 系列早期各品种

器件名称	无 ROM 型	EPROM 型	ROM/EPROM	RAM	16 位定时器	电路类型
8051	8031	(8751)	4 kB	128 字节	2	HMOSI
8051AH	8031AH	8751H	4 kB	128 字节	2	HMOSII
8052AH	8032AH	8752BH	8 kB	256 字节	3	HMOSII
80C51BH	80C31BH	87C51	4 kB	128 字节	2	CHMOS

1.3 内部结构

图 1-1 所示为 MCS-51 系列微控制器的内部功能结构。本节我们将分别讨论各专用寄存器、端口结构及操作、存储器组织、复位电路、片内振荡器和内部定时等问题。至于定时器、串行口、中断系统等部件，因其内容深广，故将另节叙述。

1.3.1 专用寄存器

专用寄存器(Special Function Registers)，简称 SFR，亦可译作特殊功能寄存器，其所在空间为片内存储器的一个区域。由图 1-2 可以看出，8051 有 21 个、8052 则有 26 个 8 位专用寄存器。括号内的专用寄存器仅 8052 才有。黑框中的三个 SFR 是 80C52 在 8052 的基础上又增加的。这些 SFR 并未占满 80H~FFH 整个地址空间，空白地址处实际在片内未安排任何寄存器。对这些地址进行读操作，通常得到一个随机数，而写操作将无结果。

1. 累加器 (Accumulator)

地址等于 E0H 的 SFR 为累加器，它是 CPU 中使用最频繁的一个寄存器。运算器 ALU 进行各种算术和逻辑运算时，大多要累加器与之配合。如图 1-1 所示，在运算前累加器中所保存的一个操作数，经暂存器 TMP2 进入 ALU 后，与从暂存器 TMP1 进入的另一个操作数在 ALU 中进行运算。所得结果往往还返送至累加器中去。累加器在代表直接地址 E0H 时，被记作 ACC，而在专指累加器的指令中，其助记符号只写作 A。

2. B 寄存器 (B Register)

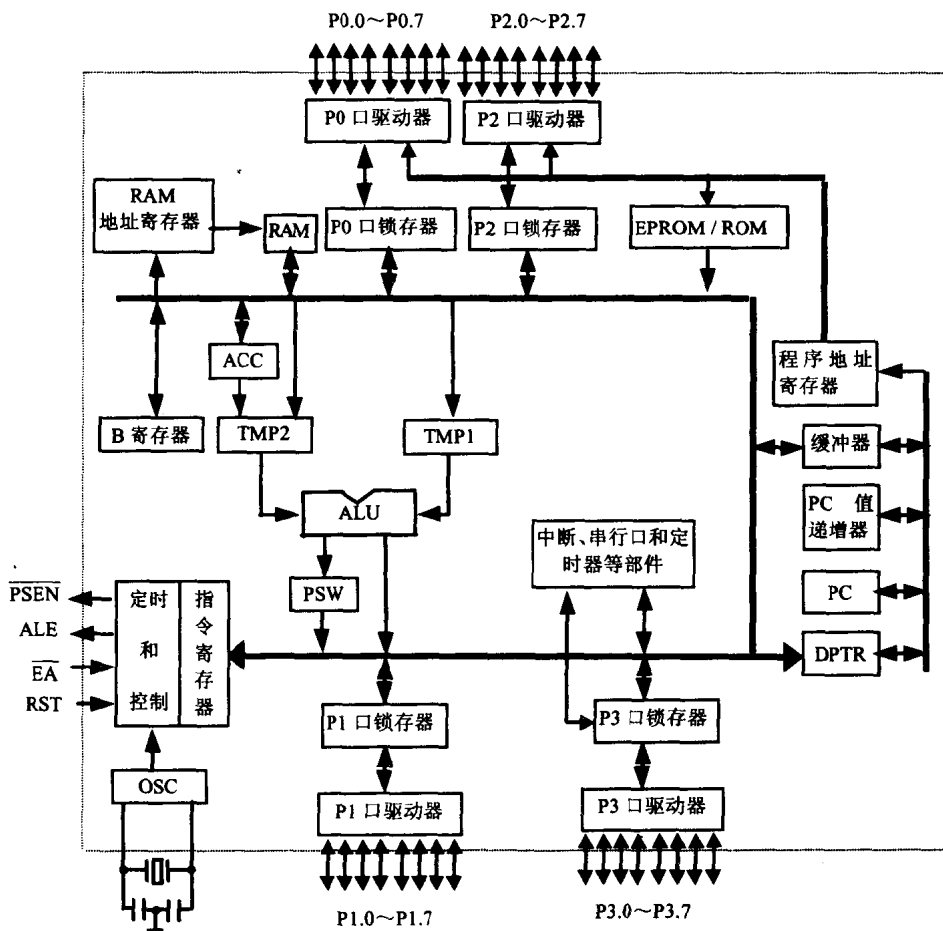


图 1-1 MCS-51 内部功能结构

执行乘除运算指令时，要用到 B 寄存器。在其他情况下，它可被用作数据寄存器，地址为 F0H。顺便指出，地址可被 8 整除的 SFR 均可按位寻址。就是说，每个字节地址包括 8 个位地址，可按这些位地址对其任何一位进行访问。图 1-2 中最左一列的 12 个 SFR，包括 B 寄存器在内，就是这样的专用寄存器。

3. 程序状态字寄存器 (Program Status Word Register)

程序状态字寄存器简写作 PSW，用来存放程序状态信息，地址为 D0H，可按位寻址。图 1-3 所示为 PSW 寄存器各位的分布情况。

CY (PSW.7)——进位标志：在加减法指令执行时，若最高位产生进位或借位，则 CY 位由硬件自动置 1，否则被清 0。此外，CY 也是布尔处理器的位累加器，参与位传送和位运

算等操作。

8 个字节							
F8H							FFH
F0H	B						F7H
E8H							EFH
E0H	ACC						E7H
D8H							DFH
D0H	PSW						D7H
C8H	(T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2)	CFH
C0H							C7H
B8H	IP	SADEN					BFH
B0H	P3						B7H
A8H	IE	SADDR					AFH
A0H	P2						A7H
98H	SCON	SBUF					9FH
90H	P1						97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	8FH
80H	P0	SP	DPL	DPH			PCON
							87H

图 1-2 SFR 分布图

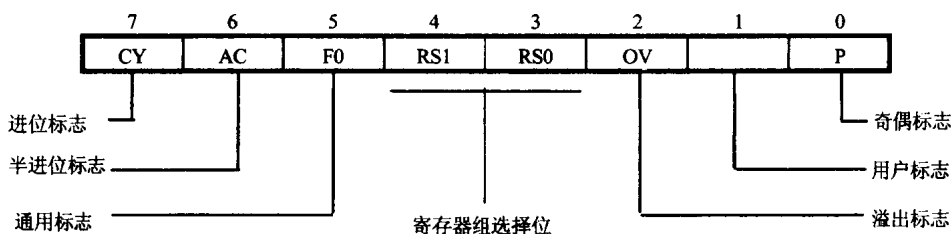


图 1-3 程序状态字寄存器 PSW

AC (PSW.6)——半进位标志：在执行加减法指令时，若低半字节向高半字节进位或借位时，AC 位被硬件自动置 1，否则被清 0。

F0 (PSW.5)——用户通用标志：由用户定义使用。

RS1 和 RS0 (PSW.4 和 PSW.3)——通用寄存器组选择位：用软件修改这两位值的组合即可对四个通用寄存器组进行选择。表 1-2 给出了 RS1 和 RS0 两位值不同组合的选择结果。复位后，这两位值为 00，本能地选中第 0 组通用寄存器为当前工作寄存器。

表 1-2 RS1 和 RS0 对通用寄存器组的选择

RS1	RS0	选中通用寄存器组	R0~R7 的地址
0	0	第 0 组	00H~07H
0	1	第 1 组	08H~0FH
1	0	第 2 组	10H~17H
1	1	第 3 组	18H~1FH

OV (PSW.2)——溢出标志：在进行带符号数的加减运算时，若运算结果超出 8 位二进制数所能容纳的范围（-128 ~ +127），则 OV 位被自动置 1，否则被清 0。

PSW.1——用户标志：由用户定义使用。

P (PSW.0)——奇偶标志：P 标志在每个指令周期中由硬件置 1 或清 0，以指示累加器中“1”的个数系偶数或奇数。在 MCS-51 中，规定用偶校验，即累加器和本标志中“1”的总数必须为偶数。

4. 堆栈指针 (Stack Pointer)

堆栈指针 SP, 地址 81H, 也是一个 8 位专用寄存器。MCS-51 的堆栈可以设置在片内 RAM 的任何一个区域, 并且向高地址方向生成。在数据压入堆栈之前, SP 之内容先自动增 1。复位后 SP 的内容为 07H, 如果不重新安排 SP 内容的话, 堆栈便从 08H 单元开始存放数据。

5. 数据指针 (Data Pointer)

数据指针 DPTR 由高字节 DPH 和低字节 DPL 两个 8 位寄存器组成, 分别占据 83H 与 82H 两地址单元。所谓数据指针, 更确切点说, 是一个对片外 RAM 进行数据存取用的地址指针。因此, 其内容是一个 16 位地址。诚然, 在 ROM 中查表和间接跳转时也要用到 DPTR。DPTR 有时以 16 位寄存器进行操作, 也有必须分成两个 8 位寄存器进行操作的场合。

6. 端口 0~3 的锁存器 (Latches of Ports 0-3)

专用寄存器空间的 80H、90H、A0H 和 B0H 等四个地址单元, 分别为端口 0~3 的四个锁存器 P0、P1、P2 和 P3。

7. 串行数据缓冲器 (Serial Data Buffer)

99H 单元为串行数据缓冲器 SBUF, 这实际上是发送缓冲器和接收缓冲器两个独立的寄存器。传送至 SBUF 的数据, 进入发送缓冲器, 并继而从那里串行发送出去; 如果对 SBUF 进行读出操作, 那么, 读之所得便是来自接收缓冲器的数据。

8. 定时器寄存器 (Timer Registers)

寄存器对 TH0、TL0(地址 8CH 和 8AH), TH1、TL1(8DH 和 8BH)与 TH2、TL2(CEH、CDH)分别为定时 / 计数器 0、1 和 2 的 16 位计数寄存器。

9. 捕捉寄存器 (Capture Registers)

寄存器对 RCAP2H、RCAP2L 系定时器 2 捕捉方式所用的捕捉寄存器。在此方式下, 一旦 8052 的 T2EX 引脚上出现有负跳变, 则 TH2 和 TL2 的内容便立即被捕获到 RCAP2H 和 RCAP2L 中去。定时器 2 亦具有初始值自动恢复运作方式, 这时捕捉寄存器用来保存原初始值。

10. 控制寄存器 (Control Registers)

中断优先级 IP、中断允许 IE、定时器方式 TMOD、定时器控制 TCON、定时器 2 控制 T2CON、串行口控制 SCON 以及电源控制 PCON 等七个专用寄存器为控制寄存器, 它们含有中断系统、定时 / 计数器和串行口的控制与状态位。这些寄存器将在后续各节中详细讨论。

11. 程序计数器 (Program Counter)

程序计数器 PC, 16 位, 其功能是保存下一次要执行的指令之地址, 每读取指令的一个字节, PC 值便自动增 1, 指向本指令的下一个字节或下一条指令的地址。这是一个用户不可访问的寄存器, 因此无地址, 亦不计入 SFR 中。

1.3.2 端口结构及运作

8051 的 32 条 I/O 线隶属于四个 8 位双向端口。每个端口均由锁存器(即专用寄存器 P0 ~ P3)、输出驱动器和输入缓冲器组成。

如图 1-4 所示, P1、P2 和 P3 等三个端口都有内部上拉电阻与其输出驱动场效应管的漏极相接; 而 P0 口则有另一场效应管接至输出驱动管的漏极。若前者截止, 则后者即为开漏输出。

1. 通用 I/O 口

上述四口的每条 I/O 线均可独立地用作输入或输出。欲使某 I/O 线作输入用, 其位锁存器必须预先置 1, 以关断输出驱动场效应管。这时, 对于 P1、P2 和 P3 三个端口来说, 该 I/O 线就会被其内部上拉电阻拉成高电平。若此脚从外部得一高电平输入, 则可读得逻辑 1; 若输入 0, 则此脚即被外来输入信号拉成低电平, 于是可读到逻辑 0。输入缓冲器接到读引脚命令后, 三态门开通, 引脚上的输入电平就会传送到内部总线上去。

倘若不预先关断输出驱动管, 外部的逻辑高电平就无法输入进来。各口的位锁存器复位后均被写成 1。如果此后写过 0, 则必须再次写 1 后方可将其相应 I/O 线置成输入方式。

作为位锁存器的 D 触发器, 当接到来自 CPU 的写锁存器信号时, 就会把内部总线上的数据锁存起来。且在写入新数据之前, 锁存器的内容不会发生变化。其输出 Q, 在来自 CPU 的读锁存器信号的作用下, 可被返送至内部数据总线。

写到位锁存器的数据何时到达输出引脚呢?

原来, 在执行改变口锁存器内容的指令过程中, 新的数值在该指令最后一个周期的 S6P2 才能到达锁存器(关于指令周期的划分, 请参见 1.3.4 节)。然而, 口锁存器实际上只有在任何时钟周期的 P1 才被其输出缓冲器采样到; P2 期间输出缓冲器将保持此前采样到的

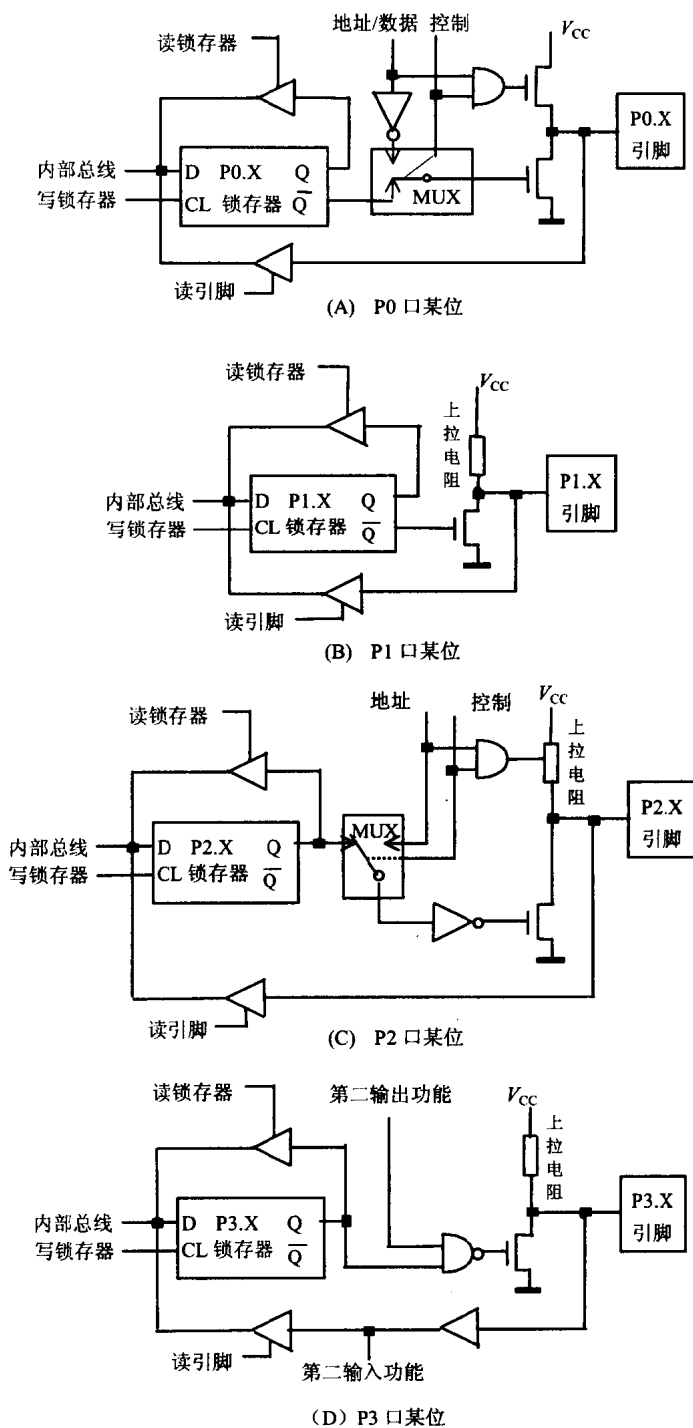


图 1-4 端口位锁存器及 I/O 缓冲器

数值。因此,新数值只有在下一个机器周期的 S1P1 时才能出现在输出引脚上。

应当指出,上述所谓内部上拉电阻,实际并非线性电阻,而是场效应晶体管。P1 和 P3 口中上拉电阻的安排如图 1-5 所示。在 MCS-51 的 HMOS 型器件中,上拉电阻始终起作用的部分为一耗尽型场效

应管,其栅极与源极相接。如果口引脚与地相接,此管可允许 0.25 mA 电流由引脚流出。由图可见,与耗尽型 FET 并联的尚有一个起低阻上拉作用的增强型场效应

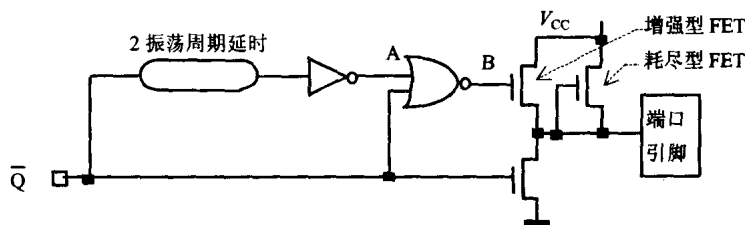


图 1-5 P1 和 P3 口的内部上拉电阻

管。后者在位锁存器之 Q 输出端发生由 0 到 1 的跳变时,在此后一机器周期的 S1P1 和 S1P2 期间,由于其栅极上有逻辑 1 电平而导通(图 1-6)。在口引脚尚未由 0 变成 1 状态前,我们可以认为该引脚是与地短接的。此时该低阻强上拉元件可允许 30 mA 电流由引脚流出。于是迅速将引脚拉成高电平,加速了跳变过程。在其关断之后,耗尽型 FET 完全可以保持住引脚的 TTL 高电平。

P2 口的情况与此相似,所不同的是,当发送的地址位为 1 时, P2 口始终使低阻上拉管处于导通状态。

在作通用 I/O 时, P0 口与其他三口的主要不同点在于无上拉电阻。更全面点说,其上拉场效应管(图 1-4 (A)),正如我们下面将要叙述的那样,只是在访问片外存储器且输出 1 时才起作用,在其他情况下是关断的。因此, P0 口某 I/O 线用作通用输出时,它是漏极开路的。

作输入时,两个场效应管都是关断的,故该引脚处于浮动状态,就是说,这是高阻输入。

因为 P1、P2 和 P3 口都具有常效内部上拉电阻,所以它们有时被称作准双向口。作输入时,有电阻起上拉作用,当引脚被从外部拉低时,这些上拉电阻又帮助提供电流 I_{IL} 。而 P0 口则被认为是真正双向口,因为作输入时该口引脚处于浮动状态。

2. 地址 / 数据总线

P0 口和 P2 口的输出驱动器以及 P0 口的输入缓冲器可用来访问片外存储器。P0 口输出片外存储器地址的低 8 位。此地址字节与读或写的数据字节分时出现在此口的 8 位 I/O 线上。故这些 I/O 线有地址 / 数据总线之称,简写作 ADDR / DATA 总线。

当片外存储器使用 16 位地址时, P2 口用来输出该地址的高 8 位,否则 P2 口之引脚将继续发出专用寄存器 P2 的内容。

由图 1-4 中可以看出, P0 口和 P2 口的输出驱动器在访问片外存储器时,在内部控制信号的作用下,可以和内部地址 / 数据和数据总线接通。如果该地址或数据位为 0,经反相后为 1,因而使输出驱动管导通,于是便从此引脚送出逻辑 0。若输出的地址或数据为 1,则该驱动管截止,但 P0 口另一个场效应管因其栅极从与门的输出端得到逻辑 1 而导通。在这种情况下,它起上拉电阻作用,结果把引脚拉成高电平,于是输出逻辑 1。这时 P0 口的输出驱动管并非漏极开路的。

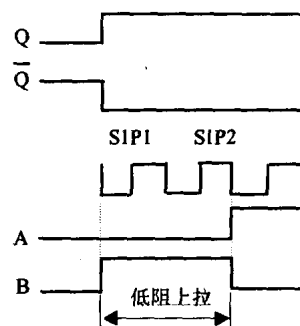


图 1-6 低阻上拉过程

P0 口作地址 / 数据总线, P2 口作地址总线时, 必须注意下列几点:

- 此时它们不能再作通用 I/O 口用;
- 访问片外存储器期间, P2 锁存器的内容不受影响, 而 P0 各位均被自动写成 1;
- 作地址 / 数据总线时, P0 口不是漏极开路的, 因此无需外接上拉电阻。

3. 引脚的复用功能

P3 口的所有引脚以及 8052 中 P1 口的两个引脚都是多功能的。它们除可用作通用 I/O 之外, 尚可作外部中断输入、计数器输入、串行口输入和输出以及对片外数据存储进行读写控制的信号传递通路。

必须指出, 复用功能(也有称之为第二功能的)只有在相应的位锁存器为 1 时才能实现。由图 1-4(D)可见, 若 Q 端为 1, 则引脚的输出电平由第二输出功能信号控制。下面让我们以 P3.7 为例来说明这个问题。

从表 1-3 可以查到, P3.7 的第二功能为外部数据存储器的读选通信号输出, 低电平有效。在执行片外存储器读指令时, P3.7 的第二输出功能信号为逻辑 0, 驱动管因其栅极从与非门之输出端得到逻辑 1 而导通。该驱动器实际上是一个低阻下拉元件, 它可以克服高阻上拉元件的作用而将引脚拉成低电平。这样便有一个有效的读选通信号送至片外 RAM 芯片, 从而完成对其进行的读操作。

至于具有复用输入功能的引脚, 如图 1-4(D)所示, 从引脚至复用功能输入线始终是有通路的。

4. 端口的负载能力和接口

作为输出引脚, 大家知道, 要接到外部负载的输入端。P1、P2 和 P3 口的每位输出驱动管可驱动 4 个 LS TTL 输入。作为输入口, 这些 HMOS 型端口可按通常方式被任一 TTL 或 NMOS 电路所驱动。HMOS 型和 CHMOS 型端口均可由集电极开路或漏极开路输出来驱动, 无需外加上拉电阻。但是要提请注意的是, 从外部输入进来的 0 到 1 的跳变和由锁存器输出的这种跳变, 两者到达引脚的过程不同。后者是通过两个振荡周期的低阻上拉作用而加速的; 前者则是只靠高阻上拉的一个相对缓慢的过程。

P0 口的输出驱动管, 每位可驱动 8 个 LS TTL 输入。但若要驱动 NMOS 输入, 需外加上拉电阻。当然本端口用作地址 / 数据总线时例外, 这在前面已有论述。

5. 读-修改-写特性

涉及到端口读操作的指令有两种, 其一读端口锁存器, 其二读端口引脚。当目的操作数为端口或口的一位时, 这类指令便是读锁存器而非读引脚之指令。例如:

ANL (逻辑与, 如 ANL P1,A);

ORL (逻辑或, 如 ORL P2,A);

XRL (逻辑异或, 如 XRL P3,A);

JBC (逢 1 清 0 并跳转, 如 JBC P1.1,LABEL);

CPL (位求反, 如 CPL P3.0);

INC (增 1, 如 INC P2);

DEC (减 1, 如 DEC P2);

DJNZ (减 1 非 0 则跳转, 如 DJNZ P1,LABEL);

MOV PX,Y,C (进位标志位送 X 口之 Y 位);

CLR PX.Y (X 口之 Y 位清 0);

表 1-3 端口引脚的复用功能

引脚	复用功能
P1.0	T2(定时/计数器 2 的外部输入)
P1.1	T2EX(定时器 2 的捕捉/重装触发引脚)
P3.0	RxD(串行口输入)
P3.1	TxD(串行口输出)
P3.2	INT0(外部中断 0 请求)
P3.3	INT1(外部中断 1 请求)
P3.4	T0(定时/计数器 0 的外部输入)
P3.5	T1(定时/计数器 1 的外部输入)
P3.6	WR(片外存储器写选通信号输出)
P3.7	RD(片外存储器读选通信号输出)

SETB PX.Y (X 口之 Y 位置 1)。

这些都是所谓读-修改-写指令。它们在执行时,先读端口锁存器之值,进行相应修改,然后再写回到端口锁存器中去。读-修改-写指令之所以要针对锁存器而非引脚,是为避免误译引脚电平之故。例如,某口一位输出用来驱动 NPN 晶体管之基极,该位写入 1 时晶体管导通。若读引脚就会读入该晶体管的基极电压,通常为 0.7 V 左右,当然被认为是逻辑 0。而读锁存器就不会发生这样的误解。

1.3.3 存储器组织

MCS-51 系列器件的程序存储器 ROM、EPROM 和数据存储器 RAM 的地址空间是分立的,其容量均可达 64 k 字节。

1. 程序存储器

如图 1-7 和图 1-8 所示,程序存储器的最低 4 kB(8052 为 8 kB) 可位于片内,亦可位于片外 ROM 中。为配合这两种情况,MCS-51 系列器件中设置了 $\overline{\text{EA}}$ (External Access) 引脚。

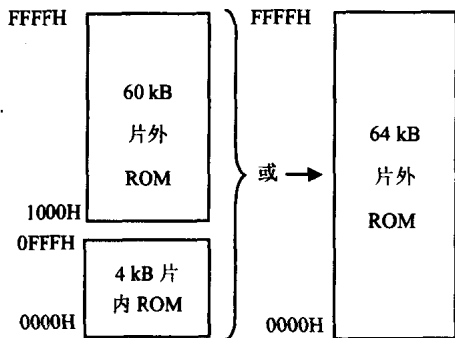


图 1-7 8051 程序存储器

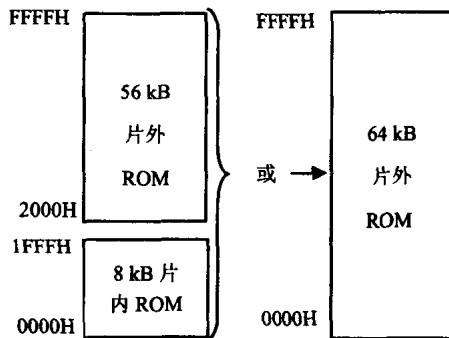


图 1-8 8052 程序存储器

在 8051 中,若 $\overline{\text{EA}}$ 接 V_{CC} , 即 +5 V 电源引脚,且 PC 的内容不大于 0FFFH,则 CPU 所执行的程序将取自片内 ROM;但若 PC 值等于或大于 1000H,即执行片外 ROM 中的程序。

在 8052 中,若 $\overline{\text{EA}}$ 接 V_{CC} ,PC 所给地址在 0000H~1FFFH 之间,则 CPU 将取片内 ROM 中的程序来执行;若地址由 2000H 至 FFFFH,则执行片外 ROM 之程序。

若 $\overline{\text{EA}}$ 接至 V_{SS} ,即接地,则全部程序将取自片外 ROM。无片内 ROM 型器件(8031 和 8032AH 等)的 $\overline{\text{EA}}$ 引脚必须外部接地,以使其全部执行片外存储器中的程序。

应当指出,上述 ROM 代表各种程序存储器,包括掩膜 ROM、EPROM 和闪电存储器等。

2. 数据存储器

如图 1-9 和图 1-10 所示,MCS-51 器件之数据存储器亦有片内与片外两个空间。MOVX 指令专用以访问片外数据存储器。片外程序存储器和片外数据存储器,根据需要也可合并成一个统一的地址空间,但这时最大容量为 64 kB,而非 128 kB。

8051 有 128 个字节的片内 RAM,21 个专用寄存器 SFR;8052 的片内 RAM 为 256 个字节,专用寄存器计 26 个。

应当注意, 8052 的专用寄存器和只能间接寻址的 RAM 区具有相同的地址 80H~FFH,

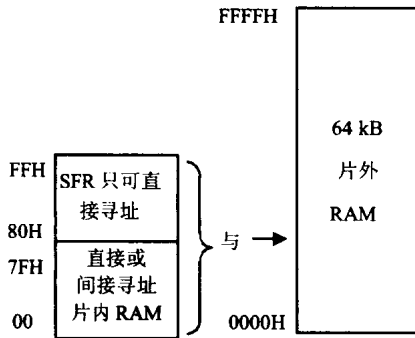


图 1-9 8051 数据存储器

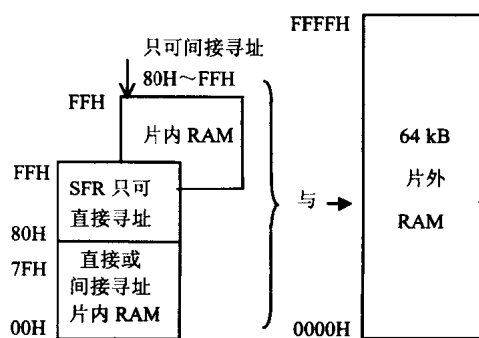


图 1-10 8052 数据存储器

但它们却是两个独立的区域, 有不同的存取方法。例如, `MOV 80H,#0AAH` 指令向专用寄存器 P0(地址为 80H)写入立即数 AAH, 而 `MOV R0,#80H` 和 `MOV @R0,#0BBH` 两条指令结合起来, 将向内部 RAM 的 80H 单元写入立即数 BBH。

在内部 RAM 中, 能被直接寻址又能被间接寻址的 128 个单元, 可以如图 1-11 所示分成三个区段。

(1) 通用寄存器组 0~3 (Register Banks 0-3): 片内 RAM 区从 00H 至 1FH 的 32 个单元为四个寄存器组, 每组包括 R0~R7 等八个通用寄存器, 也称作工作寄存器。任一时刻只有其中一组被选中, 复位后本能地选中 0 组。但用户也可通过 PSW 寄存器中的有关两位对寄存器组进行选择(见表 1-2)。这种办法给用户提供了极大的便利, 在调用子程序或中断处理时, 只需换一个寄存器组即可达到不破坏主程序所用寄存器内容之目的。当然, 返回时要注意恢复原 PSW 之值, 即恢复主程序原用的寄存器组。

在 1.3.3 之 4 中我们曾经谈到, 复位后堆栈指针 SP 的内容为 07H, 实际上, 堆栈从 08H 单元开始存放数据。由图 1-11 可见, 这恰好占用了工作寄存器组的位置。为充分利用各通用寄存器组起见, 最好把堆栈从这个区域移到地址较高的片内 RAM 区去。

(2) 可按位寻址区 (Bit Addressable Area): 20H~2FH 的 16 个地址单元系可按位寻址段, 就是说, 这 16 个字节的每一位都可被直接寻址, 它们的位地址为 00H~7FH。这些位可用两种方法表示, 汇编程序 ASM-51 都能接受。例如, 位地址 00H~07H 为字节地址 20H 之 0~7 位, 故可写成 20H.0~20H.7。当然, 本区段 16 个字节亦可按字节寻址。

← 8 个字节 →			
78H		7FH	↑
70H		77H	
68H		6FH	
60H		67H	
58H		5FH	数据存 放区段
50H		57H	
48H		4FH	
40H		47H	
38H		3FH	
30H		37H	↓
28H	40H...	...7FH	2FH ↑ 按位寻
20H	00H...	...3FH	27H ↓ 址区段
18H	3	1FH	↑
10H	2	17H	通用寄 存器组
08H	1	0FH	
00H	0	07H	↓

图 1-11 内部 RAM 三区段

(3) 数据 RAM 区 (Scratch Pad

Area): 30H~7FH 地址单元为用户便笺式数据存储区。在堆栈指针 SP 之内容被赋予本区的