

51系列单片机及 C51程序设计

王建校 杨建国 编著
宁改娣 危建国

 科学出版社

51系列单片机及C51程序设计

王建校 杨建国 编著
宁改娣 危建国

科学出版社

2002

内 容 简 介

本书首先详细介绍了 51 系列单片机的结构原理和系统设计,然后介绍了单片机调试软件 KEIL 51 的应用及其调试技巧,目的在于使读者能够边学边用,为以后的使用奠定必要的基础。同时本书还讲述了 C51 程序设计的基本语法、函数、数组和指针、结构、联合及枚举等内容,并给出大量例题,便于读者学习。

本书可作为大专院校相关专业本科生、研究生的教材或教学参考书,亦可作为培训教材。

图书在版编目(CIP)数据

51 系列单片机及 C51 程序设计/王建校等编著 —北京:科学出版社, 2002

ISBN 7-03-010266-5

I .5… II .王· III . 单片微型计算机-程序设计 IV .TP368.1

中国版本图书馆 CIP 数据核字(2002)第 014868 号

科学出版社 出版

北京牙黄城根北街16号

邮政编码 100717

[http //www sciencep com](http://www.sciencep.com)

新蕾印刷厂 印刷

科学出版社总发行 各地新华书店经销

*

2002 年 4 月第 一 版 开本 787×1092 1/16

2002 年 4 月第一次印刷 印张 19

印数 1—5 000 字数 448 000

定价:26.00 元

(如有印装质量问题,我社负责调换〈路通〉)

前 言

单片机在我国的使用已经 20 余载了,人们的应用水平也有了大幅度的提高,积累了极其丰富的经验和资料。反映这方面的书籍或教材早已琳琅满目,丰富多彩,使用时可根据需要,引用现有的技术资料 and 子程序就可以实现各种功能。

然而,随着计算机应用技术的不断发展,单片机的应用领域越来越广泛,其处理的实际问题也越来越复杂,因此,人们越来越重视单片机应用系统的研发周期和工作效率,殷切希望使用高效率的开发手段,达到事半功倍的效果。可喜的是这一步距离我们并不遥远!

以往,人们研制单片机应用系统,几乎都是使用汇编语言编写应用程序。尽管汇编语言具有能够直接操作机器硬件、指令的执行速度快等优点,但由于汇编语言不是结构化的程序设计语言,用它编写的程序可读性较差,不便于资料的交流和移植,调试也比较麻烦。而 C 语言恰好克服了汇编语言的缺点,而且又具有汇编语言的优点。本书首先向读者详细介绍 51 系列单片机的结构原理,然后以比较短的篇幅,介绍目前使用比较普遍、性能优良的单片机调试软件 KEIL 51,目的在于使读者能够边学边用,提高学习效率,同时为以后的使用奠定必要的基础,避免学用脱节。

在内容安排上,本书力求突出单片机的特点,重点解决学用结合问题。与其他同类书不同的是,本书没有介绍 8255(并行接口)、8253(定时器/计数器接口)、8250(异步通信接口)、8279(键盘显示器接口)。因为笔者认为,51 系列单片机已含有相应的功能部件或外接少许硬件电路就可以代替某些常用接口电路,因此进行了大胆的改进与取舍。另外,使用本书时,如果读者没有 C 语言基础,可以在学完第一~三章之后,根据实际需要,提前选学本书的第五~九章的内容。

考虑到学习单片机的具体情况,本书在举例时,大多同时给出了汇编语言源程序和 C 语言源程序,以便于读者学习与比较,加深对程序功能和作用的理解。根据笔者的经验,学习单片机和学习程序设计语言一样,如果不做实验,就几乎等于没有学过单片机。因此,建议读者一定要重视上机实验,千万不要忘记这一重要环节。

本书在写作过程中,曾得到申忠如教授、张彦斌教授、杨拴科副教授、张克农副教授的指点与建议,还得到宋竞梅、李峰、高歌、徐正红老师及其他老师的帮助与支持,在此一并表示诚挚的感谢。

读者如果需要本书中的原程序,可通过电子邮件与作者联系:

wjxdzx@sohu.com

作 者

2002 年 1 月于西安

目 录

第一章 单片机基础知识	1
1.1 MCS-51 单片机的特点	1
1.2 MCS-51 单片机的内部结构	2
1.2.1 中央处理器	2
1.2.2 存储器结构	5
1.2.3 片内并行接口	8
1.2.4 MCS-51 的内部资源	10
1.2.5 MCS-51 的芯片引脚	10
1.2.6 单片机的工作方式	12
1.3 定时器/计数器	15
1.3.1 定时器/计数器 0 和 1	15
1.3.2 定时器/计数器 2	17
1.3.3 定时器/计数器的控制和状态寄存器	19
1.4 串行接口	21
1.4.1 数据缓冲寄存器 SBUF	21
1.4.2 串行口控制寄存器 SCON	22
1.4.3 模式 0	23
1.4.4 模式 1	24
1.4.5 模式 2 和 3	26
1.4.6 多处理机通信	29
1.4.7 串行帧	30
1.4.8 波特率	31
1.5 中断	32
1.5.1 允许中断寄存器 IE	33
1.5.2 中断优先级寄存器 IP	33
1.5.3 优先级结构	33
1.5.4 中断响应协议	34
1.5.5 外部中断	35
1.5.6 中断请求的撤除	35
1.5.7 中断响应时间	35
1.6 MCS-51 的系统扩展	36
1.6.1 外部总线的扩展	36
1.6.2 外部程序存储器的扩展	37
1.6.3 外部数据存储器的扩展	40
1.7 MCS-51 单片机的指令系统	42
1.7.1 寻址方式	42
1.7.2 指令说明	44
1.7.3 伪指令	49
1.7.4 指令系统表	51
思考题	55
第二章 KEIL 51 应用入门	57

2.1	KEIL 51 的集成环境·····	57
2.2	简单的程序调试·····	58
2.3	建立一个项目·····	58
2.4	项目中含有多个文件·····	65
2.5	汇编语言·····	67
2.6	机器代码的效率比较·····	68
第三章	KEIL 51 软件的调试技巧 ·····	70
3.1	P1 口作为输入端口·····	70
3.2	P1 口作为输出端口·····	71
3.3	外部中断 (INT0)·····	71
3.4	定时器/计数器 0 作为定时器·····	72
3.5	定时器/计数器 0 作为计数器·····	73
3.6	调试函数·····	74
第四章	MCS-51 单片机系统设计 ·····	76
4.1	扩展并行口·····	76
4.1.1	用闲置不用的口线作为选通信号·····	76
4.1.2	部分地址译码法扩展并行输出端口·····	77
4.1.3	全地址译码法扩展并行输出端口·····	79
4.2	串并转换·····	80
4.2.1	74HC164 扩展并行输出端口·····	80
4.2.2	用 74HC165 扩展并行输入端口·····	81
4.3	静态数码管显示·····	81
4.4	动态数码管显示·····	83
4.5	专用数码管显示电路 (MAX7219)·····	86
4.6	A/D 转换 (ADC0809)·····	89
4.6.1	等待连接方式·····	90
4.6.2	中断连接方式·····	91
4.6.3	查询连接方式·····	93
4.7	A/D 转换 (TLC0831)·····	95
4.8	D/A 转换 (MAX517/518)·····	97
4.9	D/A 转换 (DAC0832)·····	101
4.10	时钟日历 (DS1302)·····	103
4.11	IC 卡 (24C01)·····	109
4.12	温度转换 (DS18B20)·····	113
4.13	SVM 系列字符型显示模块·····	119
4.13.1	概述·····	119
4.13.2	指令描述·····	121
4.13.3	时序图和硬件连接图·····	123
4.13.4	程序参考·····	125
4.14	键盘控制器 7289A·····	132
4.14.1	引脚介绍·····	132
4.14.2	控制指令·····	133
4.14.3	SPI 串行口·····	139
4.14.4	设计实例·····	140

4.14.5 接口程序	142
第五章 C语言的基本知识	149
5.1 C语言的特点及其程序结构	149
5.2 C语言的标识符和关键字	154
第六章 C51程序设计的基本语法	156
6.1 数据类型	156
6.2 常量	162
6.3 变量及其存储模式	163
6.4 用 typedef 重新定义数据类型	167
6.5 运算符与表达式	168
6.5.1 赋值运算符	168
6.5.2 算术运算符	168
6.5.3 增量和减量运算符	169
6.5.4 关系运算符	172
6.5.5 逻辑运算符	174
6.5.6 位运算符	175
6.5.7 复合赋值运算符	176
6.5.8 逗号运算符	178
6.5.9 条件运算符	179
6.5.10 指针和地址运算符	179
6.5.11 强制类型转换运算符	180
6.5.12 sizeof 运算符	181
6.6 表达式语句	183
6.7 复合语句	183
6.8 条件语句	185
6.9 开关语句	186
6.10 循环语句	188
6.10.1 while 语句	188
6.10.2 do-while 语句	189
6.10.3 for 语句	189
6.10.4 goto 语句	190
6.10.5 continue 语句	192
6.11 返回语句	193
第七章 函数	195
7.1 函数的定义	195
7.2 函数的调用	196
7.2.1 函数的调用形式	196
7.2.2 对被调用函数的说明	197
7.2.3 函数的参数和函数的返回值	199
7.2.4 实际参数的传递方式	201
7.3 函数的递归调用与再入函数	201
7.4 中断服务函数与寄存器组定义	203
7.5 变量的存储方式	205
7.5.1 局部变量与全局变量	205

7.5.2	变量的存储种类	206
7.5.3	函数的参数和局部变量的存储器模式	211
第八章	数组和指针	212
8.1	数组的定义和引用	212
8.2	字符数组	213
8.3	数组元素赋初值	214
8.4	数组作为函数的参数	215
8.5	指针的概念	217
8.5.1	指针与地址	217
8.5.2	指针变量的定义	218
8.5.3	指针变量的引用	219
8.5.4	指针变量作为函数的参数	220
8.6	数组的指针	222
8.6.1	用指针引用数组元素	222
8.6.2	字符数组指针	223
8.7	指针的地址计算	224
8.8	函数型指针	226
8.9	返回指针型数据的函数	228
8.10	指针数组与指针型指针	229
8.10.1	指针数组	229
8.10.2	指针型指针	232
8.11	抽象型指针	234
第九章	结构、联合及枚举	235
9.1	结构的概念	235
9.1.1	结构变量的定义	235
9.1.2	结构变量的引用	237
9.1.3	结构变量的初值	238
9.1.4	结构数组	239
9.2	结构型指针	240
9.2.1	结构型指针的概念	240
9.2.2	用结构型指针引用结构元素	240
9.3	结构与函数	241
9.3.1	将结构作为函数的参数	241
9.3.2	将结构型指针作为函数的参数	243
9.4	联合	245
9.4.1	联合的定义	245
9.4.2	联合变量的引用	246
9.5	枚举	247
9.5.1	枚举的定义	247
9.5.2	枚举变量的取值	248
附录	250
附录 A	51 系列单片机一览表	250
附录 B	C51 的库函数	280

第一章 单片机基础知识

1.1 MCS-51 单片机的特点

单片机 (Microcontroller, 又称微控制器) 是在一块硅片上集成了各种部件的微型计算机, 这些部件包括中央处理器 CPU、数据存储器 RAM、程序存储器 ROM、定时器/计数器和多种 I/O 接口电路。

MCS-51 单片机的基本结构如图 1-1 所示。

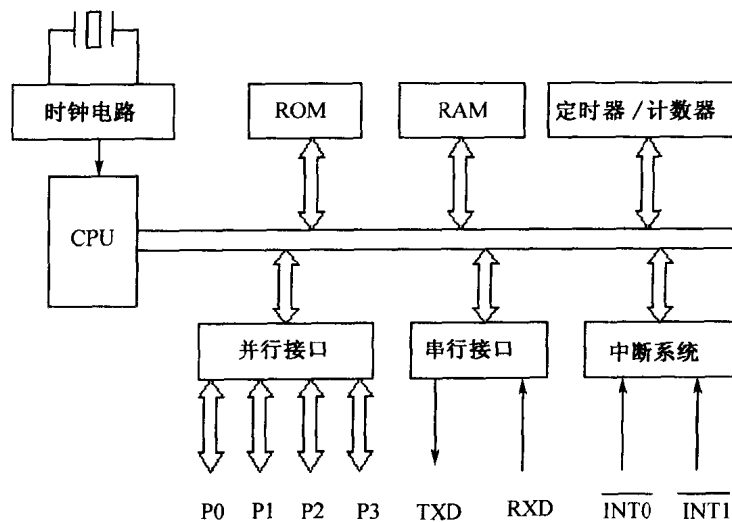


图 1-1 MCS-51 单片机的基本结构

MCS-51 系列单片机是 Intel 公司推出的通用型单片机。它的基本型产品是 8051、8031 和 8751。这 3 个产品只是片内程序存储器的制造工艺不同。8051 的片内程序存储器 ROM 为掩膜型的, 在制造芯片时已将应用程序固化进去, 使它具有了某种专用功能; 8031 片内无 ROM, 使用时需外接 ROM; 8751 片内的 ROM 是 EPROM 型的, 固化的应用程序可以方便地改写。

以上 3 个器件是 HMOS 工艺的。此外还有低功耗基本型的 COMS 工艺器件 80C51、80C31、87C51 等分别与上述器件兼容。CMOS 具有低功耗的特点, 如 8051 功耗约为 630mW, 而 80C51 的功耗只有 120mW。

除片内 ROM 类型不同外, 8051、8031、8751 的其他性能完全相同, 其结构特点如下:

- (1) 8 位 CPU;
- (2) 片内振荡器及时钟电路;

- (3) 32 根 I/O 线；
- (4) 外部存储器寻址范围 ROM、RAM 各 64KB；
- (5) 3 个 16 位的定时器/计数器；
- (6) 5 个中断源，2 个中断优先级；
- (7) 全双工串行口；
- (8) 布尔处理器。

MCS-51 系列单片机已有十多个产品，其性能如表 1-1 所示。

表 1-1 MCS-51 系列单片机性能表

ROM 形式			片内 ROM (字节)	片内 RAM (字节)	寻址范围 (字节)	I/O			中断源
片内 ROM	片内 EPROM	外接 EPROM				计数器	并行口	串行口	
8051	8751	8031	4K	128	64K	2×16	4×8	1	5
80C51	87C51	80C31	4K	128	64K	2×16	4×8	1	5
8052	8752	8032	8K	256	64K	3×16	4×8	1	6
80C252	87C252	80C252	8K	256	64K	3×16	4×8	1	7

表中列出了 4 组性能上略有差异的单片机。前两组属于同一规格，都可称为 51 系列。后两组为 52 系列，性能要高于 51 系列，除了存储器配置等差别外，8052 片内 ROM 中还掩膜了 BASIC 解释程序，因而可以直接使用 BASIC 程序。此外，87C51 和 87C52 还具有两级程序保密系统。

51 系列单片机指的是 MCS-51 系列和其他公司的 8051 衍生产品。这些衍生产品是在基本型基础上增强了各种功能的产品，如高级语言型、Flash 型、EPROM 型、A/D 型、DMA 型、多并行口型、专用接口型、双控制器串行通信型等。这些增强型的 MCS-51 系列产品，给 8 位单片机注入了新的活力，为它的开发应用开拓了更广泛的前景。Philips 公司和 Atmel 公司的 MCS-51 系列产品见附录 A。

1.2 MCS-51 单片机的内部结构

图 1-2 是 MCS-51 单片机内部结构的总框图，它可以划分为 CPU、存储器、并行口、串行口、定时器/计数器、中断逻辑几部分。

1.2.1 中央处理器

MCS-51 的中央处理器 CPU 由运算器和控制逻辑构成，其中包括若干特殊功能寄存器 (SFR)。

1. 以 ALU 为中心的运算器

算术逻辑单元 ALU 能对数据进行加、减、乘、除等算术运算，“与”、“或”、“异或”等逻辑运算以及位操作运算。

ALU 只能进行运算，运算的操作数可以事先存放到累加器 ACC 或暂存器 TMP 中，运算结果可以送回 ACC 或通用寄存器或存储单元中，累加器 ACC 也可以写为 A。B 寄存器在乘法指令中用来存放一个乘数，在除法指令中用来存放除数，运算后 B 中为部分运算结果。

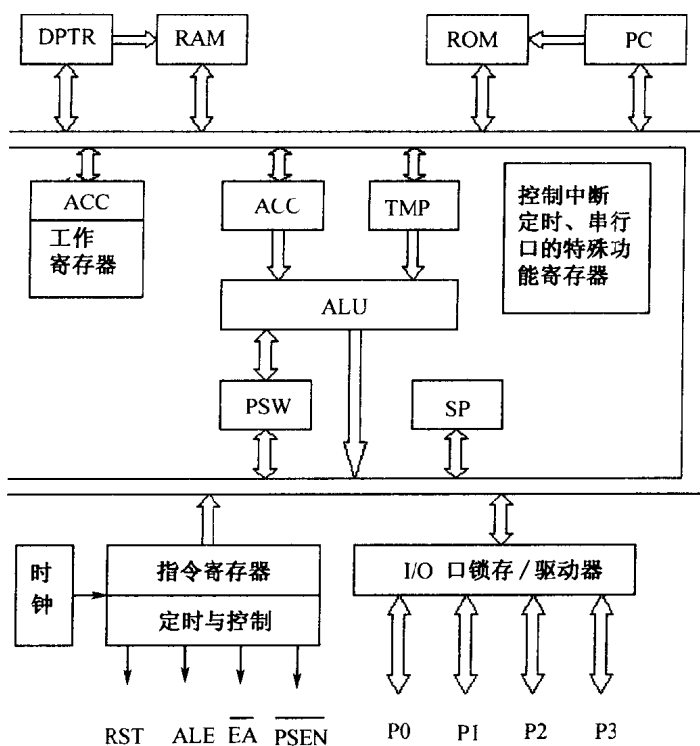


图 1-2 MCS-51 的内部结构框图

程序状态字 PSW 是个 8 位寄存器，用来寄存本次运算的特征信息，用到其中的 7 位。

D7	D6	D5	D4	D3	D2	D1	D0
CY	AC	F0	RS1	RS0	OV		P

图 1-3 PSW 的格式

PSW 的格式及各位的含义如图 1-3 所示。其中，

CY: 进位标志。有进位/借位时 CY=1，否则 CY=0。

AC: 半进位标志。当 D3 位向 D4 位产生进位/借位时 AC=1，常用于十进制调整运算中。

F0: 用户可设定的标志位，可置位/复位，也可供测试。

RS1、RS0: 4 个通用寄存器组的选择位，该两位的 4 种组合状态用来选择 0~3 寄存器组。见表 1-2。

表 1-2 RS1、RS0 与工作寄存器组的关系

RS1	RS0	工作寄存器组
0	0	0 组 (00~07)
0	1	1 组 (08~0F)
1	0	2 组 (10~17)
1	1	3 组 (18~1F)

OV: 溢出标志。当带符号数运算结果超出 $-128 \sim +127$ 范围时 $OV=1$, 否则 $OV=0$ 。当无符号数乘法结果超过 255 时, 或当无符号数除法的除数为 0 时, $OV=1$, 否则 $OV=0$ 。

P: 奇偶校验标志。每条指令执行完, 若 A 中 1 的个数为奇数时 $P=1$, 否则 $P=0$, 即奇偶校验方式。

2. 控制器、时钟电路和基本时序周期

控制逻辑主要包括定时和控制逻辑、指令寄存器、译码器以及地址指针 DPTR 和程序计数器 PC 等。

单片机是程序控制式计算机, 即它的运行过程是在程序控制下逐条执行程序指令的过程, 顺序地从程序存储器中取出指令送到指令寄存器 IR, 然后指令译码器 ID 进行译码, 译码产生一系列符合定时要求的微操作信号, 用以控制单片机各部分动作。

MCS-51 的控制器在单片机内部协调各功能部件之间的数据传送、数据运算等操作, 并对单片机发出若干控制信息。这些控制信息有的使用专门的控制线, 诸如 \overline{PSEN} 、 \overline{ALE} 、 \overline{EA} 以及 RST, 也有一些是和 P3 口的某些端子合用, 如 \overline{WR} 和 \overline{RD} 就是 P3.6 和 P3.7, 它们的具体功能在介绍 MCS-51 引脚时一起叙述。

(1) MCS-51 的时钟

时钟是时序的基础, MCS-51 片内由一个反相放大器构成振荡器, 可以由它产生时钟。时钟可以由两种方式产生: 内部方式和外部方式, 具体电路见图 1-4 (a) 和 (b)。

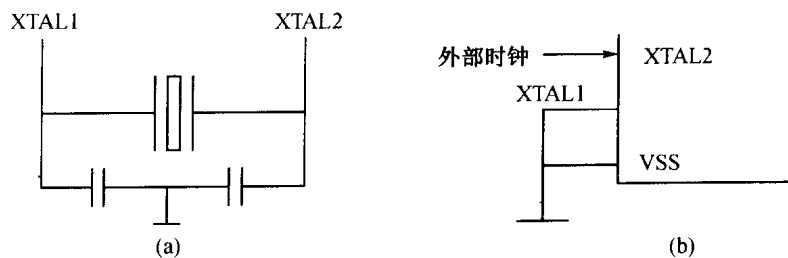


图 1-4 时钟产生电路

1) 内部方式: 在 XTAL1 和 XTAL2 端外接石英晶体作定时元件, 内部反相放大器自激振荡, 产生时钟。时钟发生器对振荡脉冲二分频, 即若石英频率 $f_{osc} = 6\text{MHz}$, 则时

钟频率 = 3MHz, 因此, 时钟是一个双相信号, 由 P1 相和 P2 相构成。 f_{osc} 可在 1.2MHz ~ 12MHz 选择, 电容可以取 30pF 左右。

2) 外部方式: 可以通过 XTAL1 和 XTAL2 接入外部时钟。

(2) MCS-51 的基本时序周期

一条指令译码产生的一系列微操作信号在时间上有严格的先后次序, 这种次序就是计算机的时序。MCS-51 的主要时序将在存储器扩展时讨论, 这里先介绍它的基本时序周期。

1) 振荡周期: 指振荡源的周期, 若为内部产生方式时, 为石英晶体的振荡周期。

2) 时钟周期 (称 S 周期): 为振荡周期的两倍, 时钟周期 = 振荡周期 P1 + 振荡周期 P2。

3) 机器周期: 一个机器周期含 6 个时钟周期 (S 周期), 即 12 个振荡周期。

4) 指令周期: 完成一条指令占用的全部时间。MCS-51 的指令周期含 1~4 个机器周期, 其中多数为单周期指令, 还有 2 周期和 4 周期指令。

若 $f_{osc} = 6\text{MHz}$, 则 MCS-51 的:

振荡周期 = $1/6\mu\text{s}$; 时钟周期 = $1/3\mu\text{s}$;

机器周期 = $2\mu\text{s}$; 指令周期 = $2\sim 8\mu\text{s}$ 。

(3) 指令部件

1) 程序计数器 PC: MCS-51 的 PC 是 16 位的计数器, 其内容为下一条待执行指令的地址, 可寻址范围 64KB。

2) 指令寄存器 IR: IR 用来存放当前正在执行的指令。

3) 指令译码器 ID: ID 对 ID 中指令操作码进行分析解释, 产生相应的控制信号。

4) 数据指针 DPTR: DPTR 是 16 位地址寄存器, 既可以用于寻址外部数据存储器, 也可以寻址外部程序存储器中的表格数据。DPTR 可以寻址 64KB 地址空间。

1.2.2 存储器结构

MCS-51 单片机的存储器结构特点之一是将程序存储器和数据存储器分开, 并有各自的寻址机构和寻址方式, 这种结构的单片机称为哈佛型结构单片机。这种结构与通用微机的存储器结构不同, 一般微机只有一个存储器逻辑空间, 可随意安排 ROM 或 RAM, 访存时用同一种指令, 这种结构称为普林斯顿型。

MCS-51 单片机在物理上有四个存储空间: 片内程序存储器和片外程序存储器、片内数据存储器和片外数据存储器。

MCS-51 片内有 256B 数据存储器 RAM 和 4KB 的程序存储器 ROM。除此以外, 还可以在片外扩展 RAM 和 ROM, 并且各有 64KB 的寻址范围。也就是最多可以在外部扩展 $2 \times 64\text{KB}$ 存储器。MCS-51 的存储器组织结构如图 1-5 所示。其中 80~FFH 单元为 8052 芯片所特有的存储区。

64K 字节的程序存储器 (ROM) 空间中, 有 4K 字节地址区对于片内 ROM 和片外 ROM 是公用的, 这 4K 字节地址为 0000H~0FFFH。而 1000H~FFFFH 地址区为外部 ROM 专用。CPU 的控制器专门提供一个控制信号 \overline{EA} 用来区分内部 ROM 和外部 ROM 的公用地址区: 当 \overline{EA} 接高电平时, 单片机从片内 ROM 的 4K 字节存储区取指令, 而

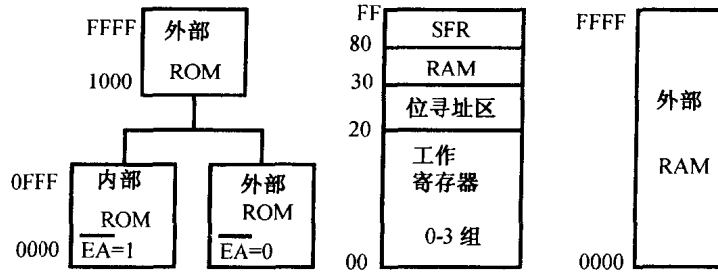


图 1-5 MCS-51 存储器组织结构

当指令地址超过 0FFFH 后，就自动地转向片外 ROM 取指令。当 \overline{EA} 接低电平时，CPU 只从片外 ROM 取指令。这种接法特别适用于采用 8031 单片机的场合，由于 8031 内部不带 ROM，所以使用时必须 $\overline{EA}=0$ ，以便直接从外部 ROM 中取指令。

程序存储器的某些单元是留给系统使用的：0000H~0002H 单元是所有执行程序的入口地址，复位以后，CPU 总是从 0000H 单元开始执行程序。0003H~0002AH 单元均匀地分为 5 段，用作 5 个中断服务程序的入口。用户程序不应进入上述区域。

数据存储 RAM 也有 64KB 寻址区，在地址上是和 ROM 重叠的。MCS-51 通过不同的信号来选通 ROM 或 RAM：当从外部 ROM 取指令时用选通信号 \overline{PSEN} ，而从外部 RAM 读写数据时采用读写信号 \overline{RD} 或 \overline{WR} 来选通。因此不会因地址重叠而出现混乱。

MCS-51 的 RAM 虽然字节数不很多，但却起着十分重要的作用。256 个字节被分为两个区域，00H~7FH 是真正的 RAM 区，可以读写各种数据，而 80H~FFH 是专门用于特殊功能寄存器 (SFR) 的区域。对于 MCS-51 安排了 21 个特殊功能寄存器，对于 8052 则安排了 26 个。每个寄存器为 8 位，所以实际上 128 个字节并没有全部利用。

对于片内 RAM 的低 128 个字节 (00H~7FH) 还可以分为三个区域。从 00H~1FH 安排了 4 组工作寄存器，每组占用 8 个 RAM 字节，记为 R0~R7。在某一时刻，CPU 只能使用其中的一组工作寄存器，工作寄存器组的选择则由程序状态寄存器 PSW 中的两位来确定。第二个区域是可按位寻址区，占用 20H~2FH，共 16 个字节 128 位。这个区域除了可以作为一般 RAM 单元进行读写外，还可以对每个字节的每一位进行操作，并且对这些位都规定了固定的位地址：从 20H 单元的第 0 位开始到 2FH 单元的第 7 位结束，共 128 位，用位地址 00H~7FH 分别与之对应。对于需要进行按位操作的数据，可以存放到这个区域。第三个区域就是一般的 RAM，地址为 30H~7FH，共 80 个字节。所以真正可以给用户使用的 RAM 单元并不多。对于 8052 芯片来说，片内多安排了 128 字节 RAM 单元，地址也为 80H~FFH，与特殊功能寄存器区域地址重叠，但在使用时，可以通过指令加以区别。

内部 RAM 的各个单元，都可以通过直接地址来寻找，而对于工作寄存器，则一般都直接用 R0~R7，对特殊功能寄存器，也是直接使用其名字较为方便。MCS-51 内部特殊功能寄存器符号及地址见表 1-3。其中带 * 号的特殊功能寄存器都是可以位寻址的，并可用“寄存器名·位”来表示，如 ACC 0, B 7 等。

表 1-3 MCS-51 特殊功能寄存器一览表

符 号	地 址	注 释
* ACC	E0H	累加器
* B	F0H	乘法寄存器
* PSW	D0H	程序状态字
SP	81H	堆栈指针
DPL	82H	数据存储器指针 (低 8 位)
DPH	83H	数据存储器指针 (高 8 位)
* IE	A8H	中断允许控制器
* IP	D8H	中断优先控制器
* P0	80H	端口 0
* P1	90H	端口 1
* P2	A0H	端口 2
* P3	B0H	端口 3
PCON	87H	电源控制及波特率选择
* SCON	98H	串行口控制器
SBUF	99H	串行数据缓冲器
* TCON	88H	定时器控制
TMOD	89H	定时器方式选择
TL0	8AH	定时器 0 低 8 位
TL1	8BH	定时器 1 低 8 位
TH0	8CH	定时器 0 高 8 位
TH1	8DH	定时器 1 高 8 位

这些寄存器分别用于以下各个功能单元：

- (1) CPU: ACC, B, PSW, SP, DPTR (由两个 8 位寄存器 DPL 和 DPH 组成)；
- (2) 并行口: P0, P1, P2, P3；
- (3) 中断系统: IE, IP；
- (4) 定时器/计数器: TMOD, TCON, T0, T1 (分别由两个 8 位寄存器 TL0 和 TH0, TL1 和 TH1 组成)；
- (5) 串行口: SCON, SBUF, PCON。

MCS-51 单片机的特殊功能寄存器中包含有堆栈指针 SP。堆栈是在内存中专门开辟出来的按照“先进后出，后进先出”原则进行存取的区域。堆栈指针 SP 就是用来指示堆栈位置的。在使用堆栈之前，先给 SP 赋值，以规定堆栈的起始位置，称为栈底。当数据存入堆栈后，堆栈指针 SP 的值也随之而变化。堆栈有两种类型：向上生长型和向下生长型，如图 1-6 所示。MCS-51 的堆栈属于向上生长型，在数据压入堆栈时，SP 的值自动加 1 作为本次进栈的地址指针，然后再存入信息。所以随着信息的存入，SP 的值越来越大。在信息从堆栈弹出之后，SP 的值随着减少。向下生长型的堆栈则相

反，栈底占用较高地址，栈顶占用较低地址。MCS-51 单片机复位后，堆栈指针 SP 总是初始化到内部 RAM 地址 07H。用户也可以根据需要通过指令改变 SP 的值，从而改变堆栈的位置。

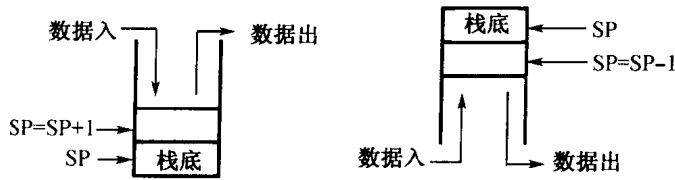


图 1-6 两种不同类型的堆栈

1.2.3 片内并行接口

MCS-51 单片机有 4 个 8 位的并行接口，记作 P0、P1、P2 和 P3，共 32 根 I/O 线，实际上它们就是 SFR 中的 4 个。每个口主要由四部分构成：端口锁存器、输入缓冲器、输出驱动器和引至芯片外的端口引脚。它们都是双向通道，每一条 I/O 线都能独立地用作输入或输出。作输出时数据可以锁存，作输入时数据可以缓冲。但这 4 个通道的功能不完全相同。图 1-7 给出了各端口位逻辑图。从图中可以看到 P0 口和 P2 口内部各有一个 2 选 1 的选择器，受内部控制信号的控制，图中的位置则是处在 I/O 口工作方式。4 个接口在进行 I/O 方式时，特性基本相同：

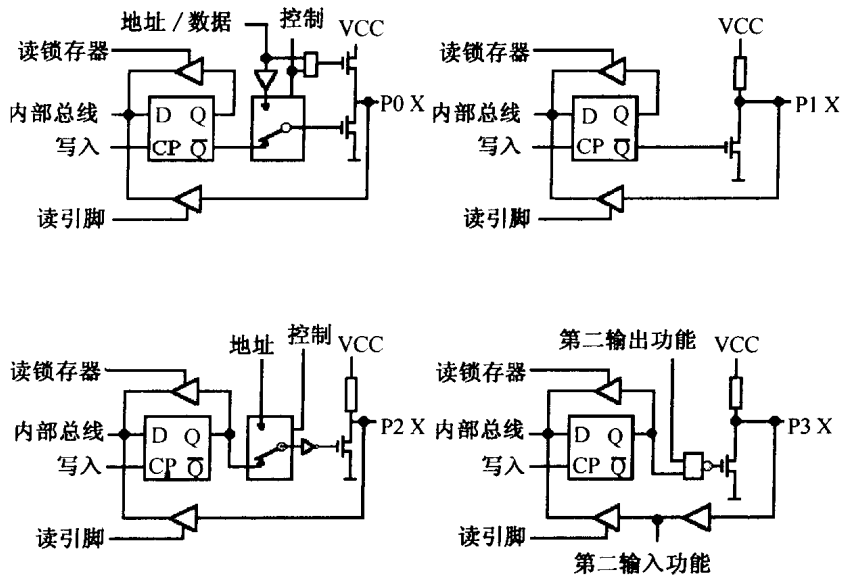


图 1-7 MCS-51 各端口位逻辑图

- (1) 作为输出口用时，内部带锁存器，故可以直接和外设相连，不必外加锁存器。
- (2) 作为输入口用时，有两种工作方式，即所谓读端口和读引脚。读端口时实际上并不从外部选入数据，而只是把端口锁存器中的内容读入到内部总线，经过某种运算和

变换后,再写回到端口锁存器。属于这类操作的指令很多,如对端口内容取反等。而读引脚时才真正地把外部的数据读入到内部总线。逻辑图中各有两个输入缓冲器,CPU根据不同的指令,分别发出“读端口”或“读引脚”信号,以完成两种不同的读操作。

(3)在端口作为外部输入线,也就是读引脚时,要先通过指令,把端口锁存器置1,然后再实行读引脚操作,否则就可能读入出错。若不先对端口置1,端口锁存器中原来状态有可能为0,加到输出驱动场效应管栅极的信号为1,该场效应管就导通,对地呈现低阻抗。这时即使引脚上输入的是1信号,也会因端口的低阻抗而使信号变低,使得外加的1信号读入后不一定是1,若先执行置1操作,则可以驱动场效应管截止,引脚信号直接加到三态缓冲器,实现正确的读入。由于在输入操作时还必须附加一个准备动作,所以这类I/O口被称为“准双向”口。

这4个接口特性上的差别主要是P0、P2和P3都还有第二功能,而P1口则只能用作I/O口。

MCS-51的芯片引脚中没有专门的地址总线 and 数据总线,在向外扩展存储器和接口时,由P2口输出地址总线的高8位A15~A8,由P0口输出地址总线的低8位A7~A0同时对P0口采用了总线复用技术,P0口又兼作8位双向数据总线D7~D0,即由P0口分时输出低8位地址或输入/输出8位数据。在不作总线扩展用时,P0口和P2口可以作为普通I/O口使用。

P0口作为低8位地址总线和8位数据总线用时,内部控制信号使MUX开关倒向上端,从而使地址/数据信号通过输出驱动器输出。当向外部存储器读写时,P0口就用作低8位地址和数据总线。这时P0口是一个真正的双向口。

P2口还可以作为高8位地址总线用,同样通过MUX开关的倒换来完成。P2在外部存储器读写时(地址大于FFH)作高8位地址线用。

P3口的每一位都有各自的第二功能,见表1-4。

表1-4 端口3的第二功能

端口位	第二功能	注 释
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0
P3.3	$\overline{\text{INT1}}$	外部中断1
P3.4	T0	计数器0计数输入
P3.5	T1	计数器1计数输入
P3.6	$\overline{\text{WR}}$	外部数据RAM写入选通信号
P3.7	$\overline{\text{RD}}$	外部数据RAM读入选通信号

4个接口的负载能力也不相同。P1、P2、P3口都能驱动4个LSTTL门,并且不需外加电阻就能直接驱动MOS电路。P0口在驱动LSTTL电路时能带八个LSTTL门,但驱动CMOS电路时若作为地址/数据总线,可以直接驱动,而作为I/O口时,需外接