



(修订版)

# 单片机的C语言 应用程序设计

马忠梅 籍顺心 张 凯 马 岩 编著



北京航空航天大学出版社

## 前 言

单片机体积小、重量轻、具有很强的灵活性而且价格不高,越来越得到广泛的应用。高速应用场合应该选用 16 位和 32 位单片机,低速应用场合仍是 4 位机的领域,其中间适用的是 8 位单片机。8051 在小到中型应用场合很常见,已成为单片机领域的实际标准。80 年代中期,Intel 公司将 8051 内核使用权以专利互换或出售形式转给世界许多著名 IC 制造厂商,如 PHILIPS、西门子、AMD、OKI、NEC、Atmel 等,这样 8051 就变成有众多制造厂商支持的、发展出上百个品种的大家族。到目前为止,其他任何一个单片机系列均未发展到如此的规模。90 年代,PHILIPS 推出支持 16 位计算的 XA 系列。XA 系列是 16 位单片机,又可完全兼容 8051 的指令系统。Intel 推出的 80C251 也是与 8051 在机器代码级兼容,这样保证了 8051 用户到 21 世纪技术的领先性。随着硬件的发展,8051 软件工具已有 C 级编译器及实时多任务操作系统(RTOS),在 RTOS 的支持下,单片机的程序设计更简单、更可靠、实时性更强。因而 8051 是单片机教学的首选机型。

C 语言是一种编译型程序设计语言,它兼顾了多种高级语言的特点,并具备汇编语言的功能。C 语言有功能丰富的库函数、运算速度快、编译效率高、有良好的可移植性,而且可以直接实现对系统硬件的控制。C 语言是一种结构化程序设计语言,它支持当前程序设计中广泛采用的由顶向下结构化程序设计技术。此外,C 语言程序具有完善的模块程序结构,从而为软件开发中采用模块化程序设计方法提供了有力的保障。因此,使用 C 语言进行程序设计已成为软件开发的一个主流。用 C 语言来编写目标系统软件,会大大缩短开发周期,且明显地增加软件的可读性,便于改进和扩充,从而研制出规模更大、性能更完备的系统。

综上所述,用 C 语言进行 8051 单片机程序设计是单片机开发与应用的必然趋势。单片机教学中的程序设计也应该以 C 语言为主,汇编语言为辅。对汇编语言掌握到只要可以读懂程序,在时间要求比较严格的模块中进行程序的优化即可。采用 C 语言也不必对单片机和硬件接口的结构有很深入的了解,编译器可以自动完成变量的存贮单元的分配,编程者就可以专注于应用软件部分的设计,大大加快了软件的开发速度。采用 C 语言可以很容易地进行单片机的程序移植工作,有利于产品中的单片机的重新选型。随着国内单片机开发工具研制水平的提高,现在的单片机仿真器普遍支持 C 语言程序的调试,为单片机编程使用 C 语言提供了便利的条件。但要注意的是断点的设置分为硬断点和软断点,只有支持硬断点的单片机仿真器才能很好地支持 C 语言的调试。另一方面,经过十余年的单片机应用推广工作,市场上和杂志上出现了一批经常与单片机配合使用的接口芯片。这些接口芯片与单片机的接口相对比较固定,接口芯片的编程也大同小异。信号处理的算法和程序已有专门的书籍。C 语言的模块化程序结构特点,可以使程序模块大家共享,不断丰富。C 语言可读性的特点,更容易使大家可以借鉴前人的开发经验,提高自己的软件设计水平。采用 C 语言,可针对单片机常用的接口芯片编制通用的驱动函数,可针对常用的功能模块、算法等编制相应的函数,这些函数经过归纳整理可形成专家库函数,供广大的单片机爱好者使用完善,这样可大大的提高国内单片机软件设计水平。国外嵌入式系统的程序设计也是采用 C 语言,我们可以借鉴它们的编程经验,进行交流,达到和国际接轨的目的。

过去长时间困扰人们的“高级语言产生代码太长,运行速度太慢不适合单片机使用”的致

命缺点已被大幅度地克服。目前,8051 上的 C 语言的代码长度,在未加入人工优化的条件下,已经做到了最优汇编程序水平的 1.2 倍~1.5 倍。可以说,比得上中等程序员的水平。目前,8051 系列芯片中,片上 ROM 空间做到 16/32K 字节的已经很多,代码效率所差的 20%~50% 已经不是重要问题。关于执行速度的问题,只要有好的仿真器的帮助,用人工优化关键代码就是很简单的事了。如果谈到开发速度、软件质量、结构严谨、程序坚固等方面的话,则 C 语言的完美绝非是汇编语言编程所能比拟的。

本次修订是为了使这本书更适合于单片机的教学使用。在原来版的基础上增添了 8051 单片机的基础知识作为本版的第一章,并对原版的各章均增加了习题。本书第一章为单片机基础知识。第二章从讨论 8051 的编程语言及其特点出发,给出了国际上现有各种 C51 编译器的性能比较,然后通过一个 Franklin/Keil C51 的编程实例讲解 C 语言的结构。第三章列举了逻辑和算术操作数,这些数据对嵌入式应用很重要。单片机有多个存储空间——程序、数据、表格等等,而且寻址不同,因此了解 8051 的各种类型的变量和各种类型的存储空间很重要。第四章讲解了分支和循环结构,它们是结构化编程方法的基础。解释了结构化编程的思想,如包括在循环开头和结尾处检测的不同。第五章为构造的数据类型,它们是函数的基础。此章包括结构、数组、指针和联合。实例包括内插比较到直接计算的查表使用。第六章阐述函数和程序结构,并有函数值的传进传出和变量的存储类型及作用域。第七章为使用多个文件开发程序进行混合编程,它是现代编程的关键。当几个编程者共同来开发一个项目时,模块化程序设计不再是因高级语言速度慢而采用汇编编程的一种技术,而是便于管理的有组织编程的关键。第八章是针对 8051 系列内部资源中断、定时器/计数器、串行口的编程及使用。这些是单片机最有特色的部分,是设计精巧系统的关键。第九章为 8051 扩展资源的编程,介绍并行接口芯片、可编程计数芯片、实时时钟/日历芯片的扩展及应用编程。第十章为输出控制通道的 D/A 变换和步进电机控制的编程。第十一章为数据采集通道的 A/D 转换及 V/F 变换的编程。第十二章为单片机的通信编程。第十三章为人机界面 LED/LCD、键盘、语音芯片、磁卡与单片机的接口及编程。对于具有实际经验的设计者或系统开发者,便会发现本书实例非常具有参考价值,由参考程序开发实用程序,将会使自己的程序更有效。对于已有其它语言的编程经验并熟悉单片机硬件的读者,通过此书的工作实例来学习一种语言会比使用其它教科书要快得多,且许多实例可直接在新的设计项目中采用。对于单片机应用课程的学生或面临设计项目的人员,本书提供了通常用于嵌入式系统的硬件及使用 8051 的实用程序,甚至一些 8051 系列应用的原理图也可以拷贝到。为了使读者掌握 8051 系列的 C 语言编程,及时建立 C51 编程调试环境,本书还在附录中列出了 Franklin C51 上机指南以及使用 EasyPack 8052F 开发装置调试 C 源程序的方法。

本书第一章、第二章及第七章至第十三章由北京理工大学计算机系马忠梅、张凯和中国科学院声学所籍顺心编写,第三章至第六章由中央民族大学物理系马岩和籍顺心编写。希望本书能对单片机的教学工作、单片机的应用推广工作起到一定的作用。由于程序和图表较多,难免有遗漏之处,恳请读者批评指正。

作者

1998 年 10 月于北京

# 目 录

<b>第一章 单片机基础知识</b>	
1.1 8051 单片机的特点 .....	(1)
1.2 8051 的内部结构 .....	(2)
1.2.1 中央处理器 .....	(3)
1.2.2 存贮器组织 .....	(5)
1.2.3 片内并行接口 .....	(7)
1.2.4 8051 的内部资源 .....	(9)
1.2.5 8051 的芯片引脚 .....	(10)
1.2.6 单片机的工作方式 .....	(10)
1.3 8051 的系统扩展 .....	(13)
1.3.1 外部总线的扩展 .....	(13)
1.3.2 外部程序存贮器的扩展 .....	(14)
1.3.3 外部数据存贮器的扩展 .....	(17)
1.4 8051 指令系统 .....	(19)
1.4.1 寻址方式 .....	(19)
1.4.2 指令说明 .....	(20)
1.4.3 伪指令 .....	(26)
1.4.4 指令系统表 .....	(27)
1.5 实用程序设计 .....	(31)
习题一 .....	(34)
<b>第二章 C 与 8051</b>	
2.1 8051 的编程语言 .....	(37)
2.2 C51 编译器 .....	(38)
2.3 Keil/Franklin C51 编程实例 .....	(39)
2.4 C51 程序结构 .....	(44)
习题二 .....	(45)
<b>第三章 C51 数据与运算</b>	
3.1 数据与数据类型 .....	(46)
3.2 常量与变量 .....	(47)
3.3 C51 数据的存贮类型与 8051 存贮器 结构 .....	(49)
3.4 8051 特殊功能寄存器(SFR)及其 C51 定义 .....	(53)
3.5 8051 并行接口及其 C51 定义 .....	(56)
3.6 位变量(BIT)及其 C51 定义 .....	(57)
3.7 C51 运算符、表达式及其规则 .....	(58)
3.7.1 C51 算术运算符及其表达式 .....	(58)
3.7.2 C51 关系运算符、表达式及优 先级 .....	(59)
3.7.3 C51 逻辑运算符、表达式及优 先级 .....	(60)
3.7.4 C51 位操作及其表达式 .....	(61)
3.7.5 自增减运算符、复合运算符及其表 达式 .....	(66)
习题三 .....	(67)
<b>第四章 C51 流程控制语句</b>	
4.1 C 语言程序的基本结构及流程图 .....	(68)
4.1.1 顺序结构及其流程图 .....	(68)
4.1.2 选择结构及其流程图 .....	(68)
4.1.3 循环结构及其流程图 .....	(69)
4.2 选择语句 .....	(71)
4.2.1 选择语句 if .....	(71)
4.2.2 switch/case 语句 .....	(73)
4.3 循环语句 .....	(74)
4.3.1 while 语句 .....	(74)
4.3.2 do—while 语句 .....	(76)
4.3.3 for 循环语句 .....	(76)
习题四 .....	(79)
<b>第五章 C51 构造数据类型</b>	
5.1 数组 .....	(80)
5.1.1 一维数组 .....	(80)
5.1.2 二维数组 .....	(83)
5.1.3 字符数组 .....	(84)
5.1.4 查表 .....	(84)
5.1.5 数组与存贮空间 .....	(85)
5.2 指针 .....	(86)
5.2.1 指针的基本概念 .....	(86)
5.2.2 数组指针和指向数组的指针变量 .....	(89)
5.2.3 指向多维数组的指针和指针变量 .....	(92)
5.2.4 关于 Franklin C51 的指针类型 .....	(93)
5.3 结构 .....	(95)
5.3.1 结构的定义和引用 .....	(95)
5.3.2 结构数组 .....	(98)
5.3.3 指向结构类型数据的指针 .....	(99)
5.4 共用体(union) .....	(101)
5.5 枚举(enum) .....	(103)
习题五 .....	(105)
<b>第六章 函 数</b>	

6.1 函数的分类 .....	(106)	10.2 12位D/A芯片AD7521 .....	(189)
6.2 函数的定义 .....	(107)	10.3 步进电机控制 .....	(192)
6.3 函数的参数和函数值 .....	(109)	习题十 .....	(197)
6.4 函数的调用 .....	(111)	<b>第十一章 8051 数据采集的C编程</b>	
6.4.1 函数调用的一般形式 .....	(111)	11.1 8位A/D芯片ADC0809 .....	(198)
6.4.2 函数调用的方式 .....	(111)	11.2 12位A/D芯片AD574 .....	(201)
6.4.3 对被调用函数的说明 .....	(111)	11.3 V/F变换芯片AD650 .....	(204)
6.4.4 函数的嵌套和递归调用 .....	(113)	11.4 频率量的测量 .....	(207)
6.4.5 函数的递归调用 .....	(113)	习题十一 .....	(211)
6.4.6 用函数指针变量调用函数 .....	(115)	<b>第十二章 8051 机间通信的C编程</b>	
6.5 数组、指针作为函数的参数 .....	(116)	12.1 点对点的串行异步通信 .....	(212)
6.5.1 用数组作为函数的参数 .....	(116)	12.2 多机通信 .....	(215)
6.5.2 用指向函数的指针变量作为函数的 参数 .....	(118)	习题十二 .....	(222)
6.5.3 用指向结构的指针变量作函数的 参数 .....	(120)	<b>第十三章 8051 人机交互的C编程</b>	
6.5.4 返回指针的函数 .....	(121)	13.1 键盘和数码显示 .....	(223)
习题六 .....	(122)	13.2 可编程键盘/显示接口芯片8279 .....	(228)
<b>第七章 模块化程序设计</b>		13.3 语音芯片TSP5220 .....	(235)
7.1 基本概念 .....	(123)	13.4 磁卡 .....	(239)
7.2 模块化程序开发过程 .....	(125)	13.5 TP $\mu$ P打印机 .....	(244)
7.3 汇编和编译 .....	(127)	13.6 字符型LCD显示模块 .....	(249)
7.4 覆盖和共享 .....	(131)	13.7 点阵型LCD显示模块 .....	(255)
7.5 库和连接/定位器 .....	(132)	习题十三 .....	(265)
7.5.1 库 .....	(132)	<b>附录A 具有51内核的单片机</b>	
7.5.2 连接/定位器 .....	(133)	表A-1 Atmel公司的8位闪速存储器单片 机 .....	(266)
7.6 混合编程 .....	(137)	表A-2 80C51微控制器系列 .....	(267)
7.7 程序优化 .....	(141)	<b>附录B Franklin C51 上机指南</b>	
习题七 .....	(142)	B.1 安装 .....	(268)
<b>第八章 8051 内部资源的C编程</b>		B.2 编译器 .....	(268)
8.1 中断 .....	(143)	B.3 连接器 .....	(279)
8.2 定时器/计数器(T/C) .....	(149)	B.4 汇编器 .....	(280)
8.3 串行口 .....	(158)	B.5 实用程序 .....	(282)
习题八 .....	(165)	<b>附录C 用Easy Pack/E 调试源程序</b>	
<b>第九章 8051 扩展资源的C编程</b>		C.1 Easy Pack/E 及 Easy Pack/SLD 简介 .....	(285)
9.1 可编程外围定时器8253 .....	(166)	C.2 Easy Pack/SLD的Windows界面 .....	(286)
9.2 可编程外围并行接口8255 .....	(171)	C.3 基本调试技巧 .....	(289)
9.3 实时时钟/日历芯片146818 .....	(176)	<b>附录D 单片机/16位微机教学实     验开发系统</b>	
9.4 可编程接口芯片8155 .....	(180)	D.1 主要功能 .....	(300)
习题九 .....	(185)	D.2 系统特点 .....	(300)
<b>第十章 8051 输出控制的C编程</b>			
10.1 8位D/A芯片DAC0832 .....	(186)		

# 第一章 单片机基础知识

## 1.1 8051 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。

8051 单片机的基本结构见图 1-1。

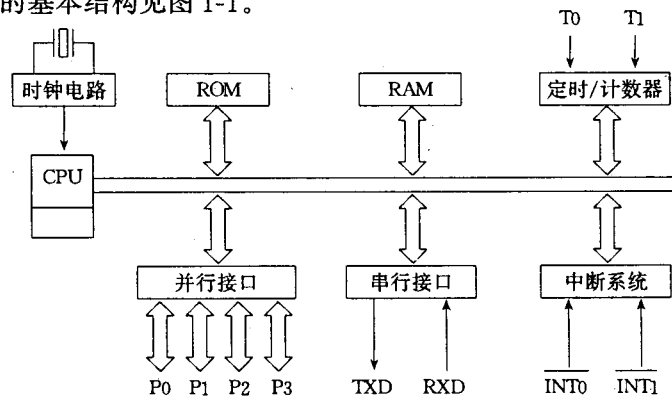


图 1-1 MCS-51 单片机的基本结构

8051 是 MCS-51 系列单片机的一个产品。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。这三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 的片内 ROM 是 EPROM 型的, 固化的应用程序可以方便地改写。

以上三个器件是 HMOS 工艺的。此外还有低功耗基本型的 CMOS 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 630 mW, 而 80C51 的功耗只有 120 mW。

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

- 8 位 CPU;
- 片内振荡器及时钟电路;
- 32 根 I/O 线;
- 外部存储器寻址范围 ROM、RAM 各 64K;
- 2 个 16 位的定时器/计数器;
- 5 个中断源, 2 个中断优先级;
- 全双工串行口;
- 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	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	2×64K	2×16	4×8	1	5
80C51	87C51	80C31	4K	128	2×64K	2×16	4×8	1	5
8052	8752	8032	8K	256	2×64K	3×16	4×8	1	6
80C252	87C252	80C232	8K	256	2×64K	3×16	4×8	1	7

表中列出了四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 87C252 还具有两级程序保密系统。

8051 单片机系列指的是 MCS-51 系列和其它公司的 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、EEPROM 型、A/D 型、DMA 型、多并行口型、专用接口型、双控制器串行通讯型等。这些增强型的 8051 系列产品,给 8 位单片机注入了新的活力,给它的开发应用开拓了更广泛的前景。Philips 公司和 Atmel 公司的 8051 系列产品见附录 A。

## 1.2 8051 的内部结构

图 1-2 是 8051 单片机片内结构的总框图,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。

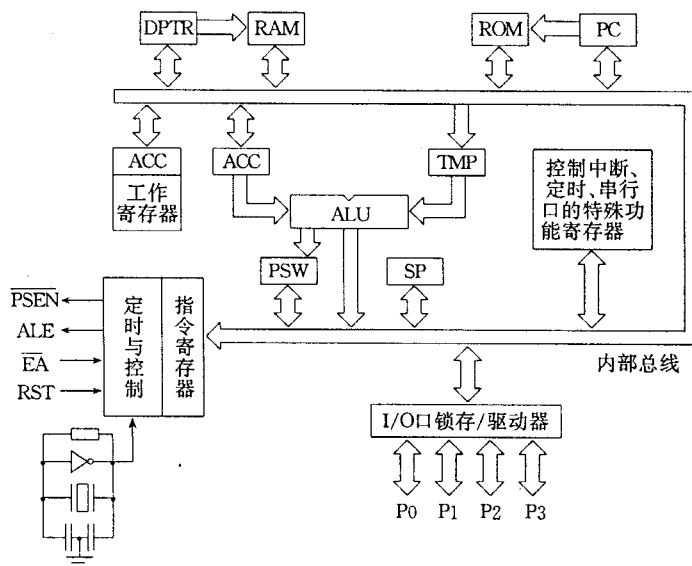


图 1-2 8051 的内部结构框图

### 1.2.1 中央处理器

8051 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)。

#### 一、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 或通用寄存器或存贮单元中,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。

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,用到其中的七位。PSW 的格式如图 1-3 所示,其各位的含义是:

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

图 1-3 PSW 的格式

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,否则 AC=0,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、RS0:四个通用寄存器组的选择位,该两位的四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(00-07H)
0	1	1 组(08-0FH)
1	0	2 组(10-17H)
1	1	3 组(18-1FH)

OV:溢出标志。当带符号数运算结果超出 -128~+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,即偶校验方式。

#### 二、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:从程序存贮器中取出指令送指令寄存器 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。

8051 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  $\overline{\text{PSEN}}$ 、 $\overline{\text{ALE}}$ 、 $\overline{\text{EA}}$  以及  $\overline{\text{RST}}$ ,也有一些是和 P3 口的某些端子合用,如  $\overline{\text{WR}}$  和  $\overline{\text{RD}}$  就是 P3.6 和 P3.7,它们的具体功能在介绍 8051 引脚时一起叙述。

### 1. 8051 的时钟

时钟是时序的基础,8051 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 的(a)和(b)。

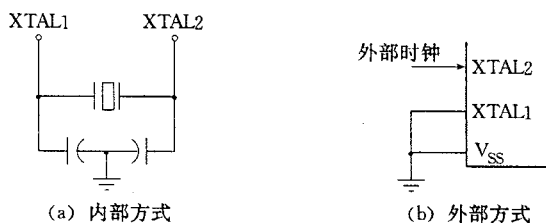


图 1-4 时钟产生电路

(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  $f_{\text{osc}}=6\text{ MHz}$ ,则时钟频率=3 MHz,因此,时钟是一个双相信号,由 P1 相和 P2 相构成。 $f_{\text{osc}}$ 可在 1.2 MHz~12 MHz 选择,小电容可以取 30 pF 左右。

(2) 外部方式:可以通过 XTAL1 和 XTAL2 接入外部时钟。

### 2. 8051 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。8051 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。

- 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。
- 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。
- 机器周期:一个机器周期含 6 个时钟周期(S 周期)。
- 指令周期:完成一条指令占用的全部时间。8051 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  $f_{\text{osc}}=6\text{ MHz}$ ,则 8051 的:

$$\begin{aligned} \text{振荡周期} &= 1/6\ \mu\text{s}; & \text{时钟周期} &= 1/3\ \mu\text{s}; \\ \text{机器周期} &= 2\ \mu\text{s}; & \text{指令周期} &= 2\sim 8\ \mu\text{s}. \end{aligned}$$

### 3. 指令部件

(1) 程序计数器 PC:8051 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 64K。

(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 IR 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 64K 地址空间。

### 1.2.2 存储器组织

8051 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。

8051 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 and 片外数据存储器。

8051 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。8051 的存储器组织结构如图 1-5 所示。其中虚线所示部分为 8052 芯片所特有的存储区。

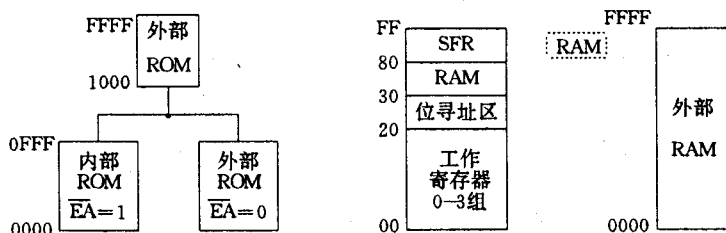


图 1-5 8051 存储器组织结构

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  $0000\text{H} \sim 0\text{FFFH}$ 。而  $1000\text{H} \sim \text{FFFFH}$  地址区为外部 ROM 专用。CPU 的控制器专门提供一个控制信号 EA 用来区分内部 ROM 和外部 ROM 的公用地址区:当 EA 接高电平时,单片机从片内 ROM 的 4K 字节存储区取指令,而当指令地址超过  $0\text{FFFH}$  后,就自动地转向片外 ROM 取指令。当 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  $\text{EA} = 0$ ,以便直接从外部 ROM 中取指令。

程序存储器的某些单元是保留给系统使用的: $0000\text{H} \sim 0002\text{H}$  单元是所有执行程序的人口地址,复位以后,CPU 总是从  $0000\text{H}$  单元开始执行程序。 $0003\text{H} \sim 0002\text{AH}$  单元均匀地分为五段,用作五个中断服务程序的入口。用户程序不应进入上述区域。

数据存储器 RAM 也有 64KB 寻址区,在地址上是和 ROM 重叠的。8051 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  $\overline{\text{PSEN}}$ ,而从外部 RAM 读写数据时采用读写信号  $\overline{\text{RD}}$  或  $\overline{\text{WR}}$  来选通。因此不会因地址重叠而出现混乱。

8051 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: $00\text{H} \sim 7\text{FH}$  是真正的 RAM 区,可以读写各种数据,而  $80\text{H} \sim \text{FFH}$  是专门用于特殊功能寄存器 (SFR) 的区域。对于 8051 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 ( $00\text{H} \sim 7\text{FH}$ ) 还可以分为三个区域。从  $00\text{H} \sim 1\text{FH}$  安排了四组工作寄存器,每组占用 8 个 RAM 字节,记为  $\text{R}0 \sim \text{R}7$ 。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 位起到 2FH 单元的第 7 位止共 128 位,用位地址 00H~FFH 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。8051 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 8051 特殊功能寄存器一览表

符 号	地 址	注 解
* 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	通道 0
* P1	90H	通道 1
* P2	A0H	通道 2
* P3	B0H	通道 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:

CPU:ACC,B,PSW,SP,DPTR(由两个 8 位寄存器 DPL 和 DPH 组成);

并行口:P0,P1,P2,P3;

中断系统:IE,IP;

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);

串行口:SCON, SBUF, PCON。

8051 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。8051 的堆栈属于向上生长型, 在数据压入堆栈时, SP 的内容自动加 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。8051 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。

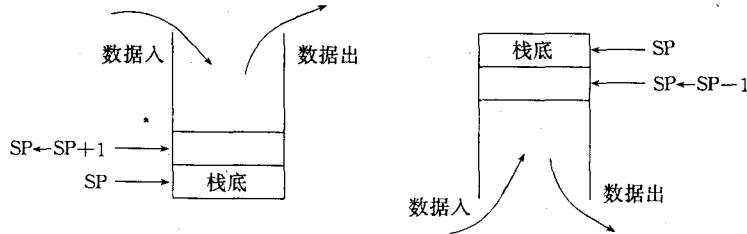


图 1-6 两种不同类型的堆栈

### 1.2.3 片内并行接口

8051 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 中的四个。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。但这四个通道的功能不完全相同。图 1-7 给出了四个通道中各个通道的一位逻辑图。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, 在如图位置则是处在 I/O 口工作方式。四个接口在进行 I/O 方式时, 特性基本相同:

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读入数据, 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。

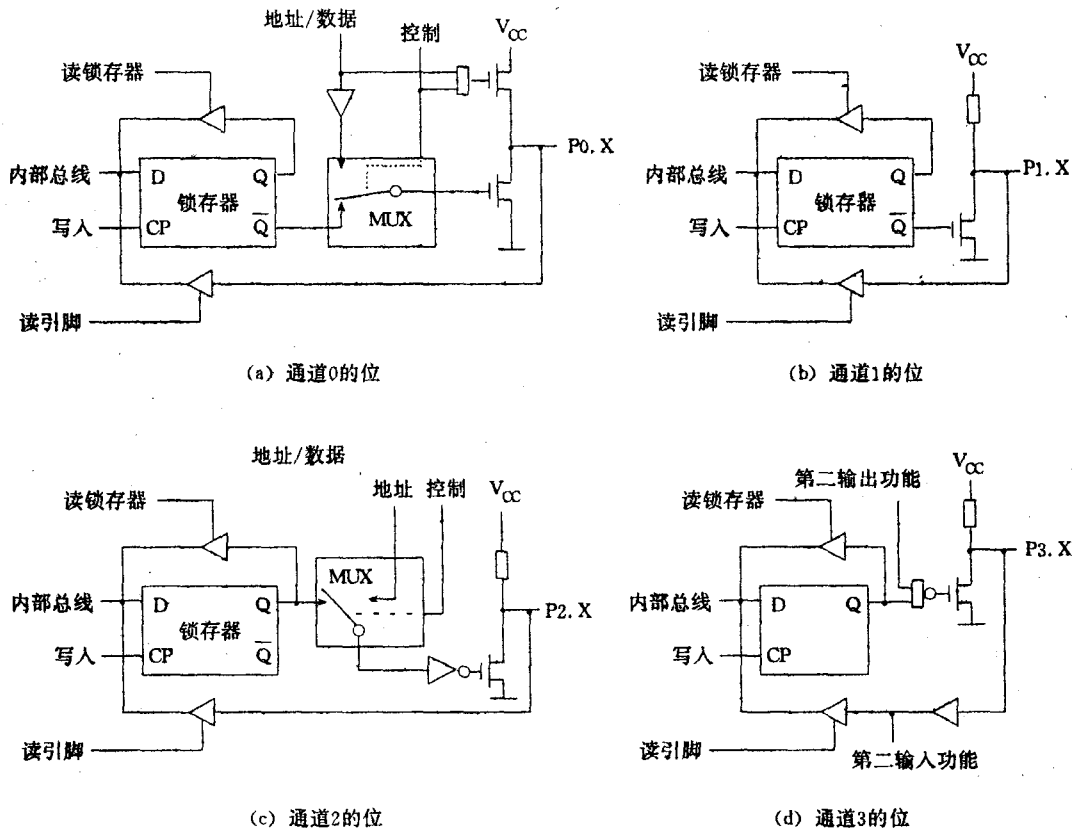


图 1-7 8051 各通道位逻辑图

这四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。

8051 的芯片引脚中没有专门的地址总线和数据总线,在向外扩展存储器 and 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 位地址和数据总线用。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。

四个接口的负载能力也不相同。P1、P2、P3 口都能驱动三个 LS TTL 门,并且不需外加电阻就能直接驱动 MOS 电路。P0 口在驱动 TTL 电路时能带八个 LS TTL 门,但驱动 MOS 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,需外接上拉电阻(电阻接 V<sub>CC</sub>),

才能驱动 MOS 电路。

表 1-4 通道 3 的第二功能

通道位	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 输入
P3.3	$\overline{\text{INT1}}$	外部中断 1 输入
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 写选通信号
P3.7	$\overline{\text{RD}}$	外部数据 RAM 读选通信号

## 1.2.4 8051 的内部资源

### 一、串行口

8051 单片机内部有一个可编程的、全双工的串行接口。串行收发存贮在特殊功能寄存器 SFR 中的串行数据缓冲器 SBUF 中的数据, SBUF 占用内部 RAM 地址 99H。但在机器内部, 实际上有两个数据缓冲器: 发送缓冲器和接收缓冲器, 因此, 可以同时保留收/发数据, 进行收/发操作, 但收/发操作都是对同一地址 99H 进行的。

### 二、定时器/计数器

8051 内部有两个 16 位可编程定时器/计数器, 记为 T0 和 T1。16 位是指它们都是由 16 个触发器构成, 故最大计数模值为  $2^{16}-1$ 。可编程是指它们的工作方式由指令来设定, 或者当计数器用, 或者当定时器用。并且计数(定时)的范围也可以由指令来设置。这种控制功能是通过定时器方式控制寄存器 TMOD 来完成的。

如果需要, 定时器在计到规定的定时值时可以向 CPU 发出中断申请, 从而完成某种定时的控制功能。在计数状态下同样也可以申请中断。定时器控制寄存器 TCON 用来负责定时器的启动、停止以及中断管理。

在定时工作时, 时钟由单片机内部提供, 即系统时钟经过 12 分频后作为定时器的时钟。计数工作时, 时钟脉冲(计数脉冲)由 T0 和 T1(即 P3.4, P3.5)输入。

### 三、中断系统

8051 的中断系统允许接受五个独立的中断源, 即两个外部中断申请, 两个定时器/计数器中断以及一个串行口中断。

外部中断申请通过  $\overline{\text{INT0}}$  和  $\overline{\text{INT1}}$  (即 P3.2 和 P3.3) 输入, 输入方式可以是电平触发(低电平有效), 也可以是边沿触发(下降沿有效)。两个定时器中断请求是当定时器溢出时向 CPU 提出的, 即当定时器由状态全 1 转为全 0 时发出的。第五个中断请求是由串行口发出的, 串行口每发送完一个数据或接收完一个数据, 就可提出一次中断申请。

8051 单片机可以设置两个中断优先级, 即高优先级和低优先级, 由中断优先控制寄存器 IP 来控制。

### 1.2.5 8051 的芯片引脚

8051 单片机是采用 40 引脚双列直插封装的芯片,有些引脚具有两种功能。引脚见图 1-8。引脚功能如下:

$V_{CC}(40)$ :电源+5 V

$V_{SS}(20)$ :接地

$XTAL1(19)$ 和  $XTAL2(18)$ :使用内部振荡电路时,用来接石英晶体和电容;使用外部时钟时,用来输入时钟脉冲。

P0 口(39~32):双向 I/O 口,既可以作地址/数据总线口,也可以作普通 I/O 口用。

P1 口(1~8):准双向通用 I/O 口。

P2 口(21~28):准双向口,既可以作地址总线口输出地址高 8 位,也可以作普通 I/O 口用。

P3 口(10~17):多用途端口,既可以作普通 I/O 口用,也可以按每位定义的第二功能操作。

$ALE/\overline{PROG}(30)$ :地址锁存信号输出端。在访问片外存储器时,ALE 为有效高电平时,P0 口输出地址低 8 位,可以用 ALE 信号做外部地址锁存器的锁存信号。 $f_{ALE}=1/6f_{osc}$ ,可以做系统中其它芯片的时钟源。

第二功能  $\overline{PROG}$  是对 8751 的 EPROM 编程时的编程脉冲输入端。

$RST/V_{PD}(9)$ :复位信号输入端。8051 接通电源后,在时钟电路作用下,该脚上出现两个机器周期(24 个振荡周期)以上的高电平,使内部复位。第二功能是  $V_{PD}$ ,即备用电源输入端。当主电源  $V_{CC}$  发生故障,降低到低电平规定值时, $V_{PD}$  将为 RAM 提供备用电源,以保证存储在 RAM 中的信号不丢失。

$\overline{EA}/V_{PP}(31)$ :内部和外部程序存储器选择线。 $\overline{EA}=0$  时访问外部 ROM 0000H~FFFFH; $\overline{EA}=1$  时,地址 0000H~0FFFH 空间访问内部 ROM,地址 1000H~FFFFH 空间访问外部 ROM。

在对 8751 的 EPROM 编程时,此脚接编程电压 12.5 V。

$\overline{PSEN}(29)$ :片外程序存储器选通信号,低电平有效。

对 8052 单片机,由于内部多一个定时器,还需要附加别的输入端,为此,又借用 P1.0 和 P1.1 作为定时器 2 的输入 T2 和 T2EX。

### 1.2.6 单片机的工作方式

单片机的工作方式包括:复位方式、程序执行方式、单步执行方式、低功耗操作方式以及 EPROM 编程和校验方式。

#### 一、复位方式

$RST$  引脚是复位信号的输入端。复位信号是高电平有效。高电平有效的持续时间应为 24

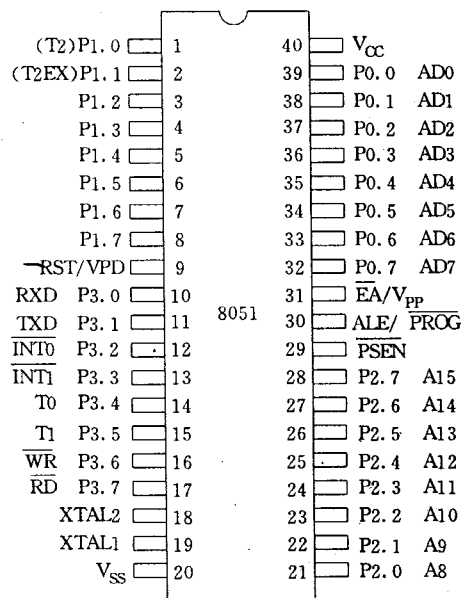


图1-8 MCS-51引脚图

个振荡周期以上。若时钟频率为 6 MHz, 则复位信号至少应持续 4 微秒以上, 才可以使单片机复位。复位以后, 07H 写入栈指针 SP, P0 口~P3 口均置 1(允许输入), 程序计数器 PC 和其它特殊功能寄存器 SFR 全部清“0”。只要该脚保持高电平, 8051 便循环复位。当 RST 端由高变低后, 8051 由 ROM 的 0000H 开始执行程序。8051 的复位操作不影响内部 RAM 的内容。当  $V_{CC}$  加电后, RAM 的内容是随机的。

单片机的复位方式有上电自动复位和手工复位两种, 图 1-9(a) 为上电复位电路, (b) 为按钮复位电路, (b) 中  $C=22\ \mu\text{F}$ ,  $R=200\ \Omega$ ,  $R_K=1\ \text{k}\Omega$ 。

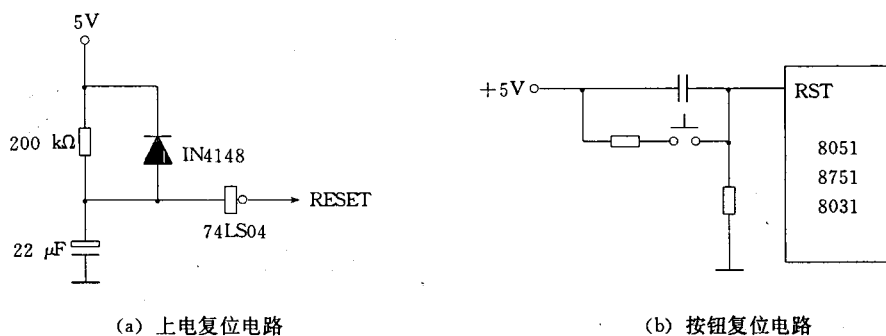


图 1-9 两种复位电路

只要  $V_{CC}$  上升时间不超过 1 ms, 通过在  $V_{CC}$  和 RST 引脚之间加一个  $10\ \mu\text{F}$  的电容器, 就可以实现自动上电复位, 即打开电源就可以自动复位。

## 二、程序执行方式

程序执行方式是单片机的基本工作方式。所执行的程序可以在内部 ROM、外部 ROM 或者同时放在内外 ROM 中。若程序放在外部 ROM 中(如对 8031), 则应使  $\overline{EA}=0$ ; 否则, 可令  $\overline{EA}=1$ 。由于复位之后  $PC=0000\text{H}$ , 所以程序的执行总是从地址 0000H 开始的。但真正的程序一般不可能从 0000H 开始存放, 因此, 需要在 0000H 单元开始存放一条转移指令, 从而使程序跳转到真正的程序入口地址。

## 三、单步执行方式

单步执行方式是使程序的执行处于外加脉冲(通常用一个按键产生)的控制下, 一条指令一条指令地执行, 即按一次键, 执行一条指令。

单步执行方式可以利用 8051 的中断控制来实现。其中断系统规定: 从中断服务程序返回以后至少要执行一条指令后才能重新进入中断。将外加脉冲加到  $\overline{INT0}$  输入, 平时为低电平。通过编程规定  $\overline{INT0}$  信号是低电平有效, 因此不来脉冲时总是处于响应中断的状态。在中断服务程序中要安排这样的指令:

```
JNB      P3.2, $      ;若 $\overline{INT0}=0$ , 不往下执行
JB       P3.2, $      ;若 $\overline{INT0}=1$ , 不往下执行
RETI                               ;返回主程序执行一条指令
```

因此, 只有  $\overline{INT0}$  上来一个正脉冲, 才能通过第一、第二两条指令, 返回主程序并执行一条指令。由于  $\overline{INT0}$  此时已回到 0, 故重新进入中断, 在第一条指令处等待正脉冲的到来。从而实现来一个脉冲执行一条指令的单步操作。

#### 四、低功耗操作方式

CMOS 型单片机有两种低功耗操作方式:节电操作方式和掉电操作方式。在节电方式时,CPU 停止工作,而 RAM、定时器、串行口和中断系统继续工作。在掉电方式时,仅给片内 RAM 供电,片内所有其它的电路均不工作。

CMOS 型单片机用软件来选择操作方式,由电源控制寄存器 PCON 中的有关位控制。这些有关的位是:

IDL (PCON. 0)	;节电方式位。IDL=1 时,激活节电方式
PD (PCON. 1)	;掉电方式位。PD=1 时,激活掉电方式
GF0 (PCON. 2)	;通用标志位
GF1 (PCON. 3)	;通用标志位

##### 1. 节电方式

一条将 IDL 位置 1 的指令执行后,80C51 就进入节电方式。这时提供给 CPU 的时钟信号被切断,但时钟信号仍提供给 RAM、定时器、中断系统和串行口,同时 CPU 的状态被保留起来,也就是栈指针 SP、程序计数器 PC、程序状态字 PSW、累加器 ACC 及通用寄存器的内容。在节电方式下, $V_{CC}$  仍为 5V,但消耗电流由正常工作方式的 24 mA 降为 3.7 mA。

可以有两途径退出节电方式恢复到正常方式。

一种途径是有任一种中断被激活,此时 IDL 位将被硬件清除,随之节电状态被结束。中断返回时将回到进入节电方式的指令后的一条指令,恢复到正常方式。

PCON 中的标志位 GF0 和 GF1 可以用作软件标志,若置 IDL=1 的同时也置 GF0/GF1=1,则节电方式中激活的中断服务程序查询到此标志便可以确定服务的性质。

退出节电方式的另一种方法是靠硬件复位,复位后 PCON 中各位均被清“0”。

##### 2. 掉电方式

一条将 PD 位置 1 的指令执行后,80C51 就进入掉电工作方式。掉电后,片内振荡器停止工作,时钟冻结,一切工作都停止,只有片内 RAM 的内容被保持,SFR 内容也被破坏。掉电方式下  $V_{CC}$  可以降到 2 V,耗电仅 50  $\mu$ A。

退出掉电方式恢复正常工作方式的唯一途径是硬件复位。应在  $V_{CC}$  恢复到正常值后再进行复位,复位时间需 10 ms 时间,以保证振荡器再启动并达到稳定,实际上复位本身只需 25 个振荡周期(2  $\mu$ s~4  $\mu$ s)。但在进入掉电方式前, $V_{CC}$  不能掉下来,因此要有掉电检测电路。

#### 五、EPROM 编程和校验方式

对于内部集成有 EPROM 的 8051 单片机,可以进入编程或校验方式。

##### 1. 内部 EPROM 编程

编程时,时钟频率应在 4 MHz~6 MHz 的范围内,其余有关引脚的接法和用法如下:

P1 口和 P2 口的 P2.0~P2.3 为 EPROM 的 4K 的高地址输入,P1 口为低 8 位地址;

P2.4~P2.6 以及  $\overline{PSEN}$  应为低电平;

P0 口为编程数据输入;

P2.7 和 RST 应为高电平,RST 的高电平可为 2.5 V,其余的都以 TTL 的高低电平为准;

$\overline{EA}/V_{PP}$  端加 +12.5 V 的编程脉冲,此电压要求稳定,不能大于 12.5 V,否则会破坏 EPROM;

在  $\overline{EA}/V_{PP}$  出现正脉冲期间, $\overline{ALE}/\overline{PROG}$  端上加 50 ms 的负脉冲,完成一次写入。