



普通高等教育“十三五”规划教材
电子信息与电气学科规划教材
全国电子信息类优秀教材

单片机原理实用教程 ——基于 Proteus 虚拟仿真 (第4版)

◆ 徐爱钧 罗明璋 熊晓东 徐 阳 编著

Electronic Information
and Electrical



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

五”规划教材
规划教材
教材

单片机原理实用教程 ——基于 Proteus 虚拟仿真 (第 4 版)

徐爱钧 罗明璋 熊晓东 徐 阳 编著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是电子信息与电气学科规划教材,并获得全国电子信息类优秀教材奖。本书系统地阐述基于 Proteus 虚拟仿真技术的 8051 单片机原理与接口技术,包括 8051 单片机基本组成、中断系统、定时器/计数器、串行口,8051 指令系统,汇编语言及高级语言 Keil C51 应用程序设计技术,键盘与显示器接口技术,DAC 及 ADC 接口技术等。本书给出汇编语言和 C 语言编程范例,以及相应的 Proteus 虚拟仿真电路图,以实例方式介绍基于 Proteus 平台进行单片机应用系统设计的方法,还介绍了具有在线仿真调试功能的 STC15 系列新型 Flash 单片机的功能特点与应用方法。

本书可作为高等学校自动化、电子信息工程、计算机应用等相关专业单片机原理与应用课程的教材,也可供广大从事单片机应用系统开发的工程技术人员阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

单片机原理实用教程:基于 Proteus 虚拟仿真/徐爱钧等编著. — 4 版. — 北京:电子工业出版社,2018.6
ISBN 978-7-121-34695-8

I. ①单… II. ①徐… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2018)第 147678 号

策划编辑:戴晨辰

责任编辑:戴晨辰

印 刷:北京京师印务有限公司

装 订:北京京师印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1092 1/16 印张:23.25 字数:670 千字

版 次:2009 年 1 月第 1 版

2018 年 6 月第 4 版

印 次:2018 年 6 月第 1 次印刷

定 价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: dcc@phei.com.cn。

前 言

随着单片机应用技术的普及,希望学习单片机的人越来越多。在全国高等教育的工科院校中,已普遍开设单片机原理与应用相关课程。由于单片机本身的特点,传统教学方法很难在教学中体现单片机的实际运行过程,尤其是一些涉及硬件的操作,如定时器/计数器控制、外围功能接口设计等,仅在课堂上空对空的讲述很难让学生理解,教学效果欠佳。Proteus 软件的出现带来了契机,利用 Proteus 虚拟仿真技术,可以通过 PC 现场绘制原理图,并直接在原理图上编写调试应用程序,配合各种虚拟仪表来展现整个单片机系统的运行过程,很好地解决了长期以来困扰单片机教学中软件和硬件无法很好结合的难题,使单片机的学习过程变得形象直观。

早期单片机应用开发大多采用汇编语言编程。汇编语言是一种直接针对硬件的机器语言,其编程效率不高,程序不易移植和维护。现在已经普遍采用 C 语言进行单片机应用编程,C 语言具有类似自然语言的特点,它既能直接操作机器硬件,又可以提高编程效率。德国 Keil 公司推出的 C51 被公认为是一个很有效的单片机 C 语言编程工具。Proteus 虚拟硬件平台可以与 Keil C51 完美结合,在原理图中直接调入 C 语言编写的应用程序,进行源代码仿真调试,实现对系统性能的综合评估,验证各项技术指标。

深圳宏晶科技有限公司研发的 STC 系列增强型 8051 单片机,是中国本土单片机领域内的佼佼者,其推出的 STC15 系列单片机具有 ISP(在系统编程)、IAP(在应用编程)和在线仿真调试功能,为单片机的学习和开发应用提供了更为方便的手段。结合 Proteus 软件平台,在成功进行虚拟仿真并获得期望结果的条件下,再利用 STC15 系列单片机制作实际硬件系统,进行在线调试,可以获得事半功倍的效果。

本书是电子信息与电气学科规划教材,并获得全国电子信息类优秀教材奖。本书在第 3 版的基础上进行了修订,在构思及选材上,注意尽量符合单片机应用系统的发展要求,增加了 Keil C51 应用程序设计的内容,给出了大量 Proteus 仿真设计实例,并同时给出了汇编语言和 C 语言程序代码,方便读者对照学习。

全书共分为 11 章。

第 1 章阐述 8051 单片机的基本组成,包括 8051 单片机的特点与基本结构、存储器结构、CPU 时序、复位信号与复位电路、并行 I/O 口结构,还介绍了深圳宏晶科技有限公司推出的 STC 系列新型 8051 单片机。

第 2 章阐述 Proteus 虚拟仿真技术,介绍在 ISIS 集成环境中绘制原理电路图、汇编语言及 C 语言源代码调试方法,以及与 Keil 环境联机仿真。

第 3 章阐述 8051 单片机的指令系统与汇编语言程序设计。

第 4 章阐述 Keil C51 应用程序设计,介绍了 C51 的基本语句、数据类型、Keil C51 对 ANSI C 的扩展,以及库函数等。

第 5 章~第 7 章分别阐述 8051 单片机的中断系统、定时器/计数器及串行口的工作原理与应用方法。

第 8 章阐述 8051 单片机系统扩展的方法。

第9章阐述键盘与显示器接口技术,介绍LED/LCD显示器、矩阵键盘与单片机的接口方法。

第10章阐述DAC及ADC接口技术,介绍传统并行接口及新型串行接口DAC、ADC与单片机的接口方法。

第11章给出6个完整的Proteus虚拟仿真设计实例及其汇编和C51源程序。

本书配有Proteus设计范例,通过学习,读者可以较快地掌握单片机基本原理及接口应用技术,为实际工作打下坚实的基础。

本书包含配套教学资源包,读者可登录华信教育资源网(www.hxedu.com.cn),注册后免费下载。为帮助读者更好地理解所讲内容,本书还制作了部分例子的虚拟仿真演示视频,读者可扫描二维码观看,程序源代码可扫描二维码下载。



虚拟仿真演示视频



程序源代码

在本书修订过程中,熊晓东负责第1章~第2章的编写,徐阳负责第3章~第4章的编写,罗明璋负责第5章~第8章的编写,彭秀华、刘冰、杨青胜、肖恩凯、郑鹏鹏、黄存坚、范林、方小玲、杨新浩、牛宣云、郑玉章等参加了其他部分章节的编写和程序调试工作,全书由徐爱钧负责完成统稿。编写过程中得到广州风标电子有限公司(Proteus中国大陆总代理)匡载华总经理的大力支持和热情帮助,在此一并表示感谢。由于作者水平有限,书中难免有错误和不妥之处,恳请广大读者批评指正。Proteus的DEMO软件可在Proteus官方网站(www.labcenter.co.uk)下载,或者与广州风标电子有限公司联系。

编著者 于长江大学

2018年6月

目 录

第 1 章 8051 单片机的基本组成	1
1.1 8051 单片机的特点与基本结构	1
1.2 8051 单片机的存储器结构	5
1.3 CPU 时序	7
1.4 复位信号与复位电路	9
1.5 并行 I/O 口结构	10
1.6 STC 系列新型 8051 单片机简介	12
复习思考题 1	16
第 2 章 Proteus 虚拟仿真	17
2.1 集成环境 ISIS	17
2.2 绘制原理图	19
2.3 创建汇编语言源代码仿真文件	21
2.4 在原理图中进行源代码仿真调试	23
2.5 原理图与 Keil 环境联机仿真	25
复习思考题 2	30
第 3 章 指令系统与汇编语言程序设计	31
3.1 指令助记符和字节数	31
3.2 寻址方式	32
3.2.1 寄存器寻址	32
3.2.2 直接寻址	33
3.2.3 立即寻址	33
3.2.4 寄存器间接寻址	34
3.2.5 变址寻址	34
3.2.6 相对寻址	35
3.2.7 位寻址	35
3.3 指令分类详解	36
3.3.1 算术运算指令	36
3.3.2 逻辑运算指令	38
3.3.3 数据传送指令	39
3.3.4 控制转移指令	40
3.3.5 位操作指令	43
3.4 汇编语言程序格式与伪指令	43
3.5 应用程序设计	45
3.6 定点数运算符程序	47
复习思考题 3	57
第 4 章 Keil C51 应用程序设计	59
4.1 Keil C51 程序设计的基本语法	59
4.1.1 Keil C51 程序的一般结构	59
4.1.2 数据类型	60
4.1.3 常量、变量及其存储模式	61
4.1.4 运算符与表达式	62
4.2 C51 程序的基本语句	66
4.2.1 表达式语句	66
4.2.2 复合语句	66
4.2.3 条件语句	66
4.2.4 开关语句	67
4.2.5 循环语句	67
4.2.6 goto、break、continue 语句	68
4.2.7 返回语句	69
4.3 函数	69
4.3.1 函数的定义与调用	69
4.3.2 中断服务函数与寄存器组定义	70
4.4 Keil C51 编译器对 ANSI C 的扩展	71
4.4.1 存储器类型与编译模式	71
4.4.2 关于 bit、sbit、sfr 和 sfr16 数据类型	72
4.4.3 一般指针与基于存储器的指针及其转换	75
4.4.4 C51 编译器对 ANSI C 函数定义的扩展	76
4.5 C51 编译器的数据调用协议	79
4.5.1 数据在内存中的存储格式	79
4.5.2 目标代码的段管理	81
4.6 与汇编语言程序的接口	83

4.6.1 参数传递规则	83	7.3.3 单片机与 PC 之间的通信	151
4.6.2 C51 与汇编语言混合编程 举例	87	复习思考题 7	154
4.7 绝对地址访问	89	第 8 章 8051 单片机系统扩展	155
4.7.1 采用扩展关键字_at_或指针 定义变量的绝对地址	89	8.1 程序存储器扩展	155
4.7.2 采用预定义宏指定变量的绝对 地址	90	8.2 数据存储器扩展	156
4.8 Keil C51 库函数	91	8.3 并行 I/O 口扩展	158
4.8.1 本征库函数	91	8.4 利用 I ² C 总线进行系统扩展	170
4.8.2 字符判断转换库函数	91	8.5 8051 单片机的节电工作方式	184
4.8.3 输入/输出库函数	92	8.5.1 空闲和掉电工作方式	185
4.8.4 字符串处理库函数	93	8.5.2 节电方式的应用	186
4.8.5 类型转换及内存分配库函数	94	复习思考题 8	187
4.8.6 数学计算库函数	95	第 9 章 键盘与显示器接口技术	188
复习思考题 4	95	9.1 LED 显示器接口技术	188
第 5 章 中断系统	97	9.1.1 七段 LED 数码管显示器	188
5.1 中断的概念	97	9.1.2 串行接口 8 位共阴极 LED 驱动器 MAX7219	194
5.2 中断系统的结构与中断控制	98	9.2 键盘接口技术	200
5.3 中断响应	101	9.2.1 编码键盘接口技术	201
5.4 中断系统应用举例	103	9.2.2 非编码键盘接口技术	204
5.4.1 中断源扩展	103	9.2.3 键值分析	210
5.4.2 中断嵌套	105	9.3 8279 可编程键盘/显示器芯片 接口技术	213
复习思考题 5	108	9.3.1 8279 的工作原理	213
第 6 章 定时器/计数器	109	9.3.2 8279 的数据输入、显示输出 及命令格式	215
6.1 定时器/计数器的工作方式与 控制	109	9.3.3 8279 的接口方法	220
6.2 定时器/计数器应用举例	112	9.4 LCD 接口技术	225
6.2.1 初值和最大定时时间计算	112	9.4.1 LCD 的工作原理和驱动方式	226
6.2.2 定时器方式应用	113	9.4.2 点阵字符液晶显示模块接口 技术	226
6.2.3 计数器方式应用	120	9.4.3 点阵图形液晶显示模块接口 技术	242
6.3 利用定时器产生音乐	122	复习思考题 9	252
复习思考题 6	128	第 10 章 DAC 及 ADC 接口技术	253
第 7 章 串行口	129	10.1 ADC 及 DAC 的主要技术 指标	253
7.1 串行通信方式	129	10.2 DAC 接口技术	254
7.2 串行口的工作方式与控制	130	10.2.1 无内部锁存器的 DAC 接口 方法	255
7.3 串行口应用举例	134		
7.3.1 串口/并口转换	134		
7.3.2 单片机之间的通信	137		

10.2.2	DAC0832 与 8051 单片机的 接口方法	256	11.2.2	硬件电路设计	304
10.2.3	DAC1208 与 8051 单片机的 接口方法	261	11.2.3	软件程序设计	307
10.2.4	串行 DAC 与 8051 单片机的 接口方法	263	11.3	单片机 SD 卡读/写接口设计	315
10.2.5	利用 DAC 接口实现波形 发生器	270	11.3.1	功能要求	315
10.3	ADC 接口技术	277	11.3.2	硬件电路设计	315
10.3.1	比较式 ADC0809 与 8051 单片机的接口方法	278	11.3.3	软件程序设计	318
10.3.2	积分式 ADC7135 与 8051 单片机的接口方法	283	11.4	电子万年历设计	322
10.3.3	串行 ADC 与 8051 单片机的 接口方法	291	11.4.1	功能要求	322
复习思考题 10		295	11.4.2	硬件电路设计	322
第 11 章	Proteus 虚拟仿真设计实例	297	11.4.3	软件程序设计	324
11.1	红外遥控系统设计	297	11.5	电子密码锁设计	335
11.1.1	功能要求	297	11.5.1	功能要求	335
11.1.2	硬件电路设计	297	11.5.2	硬件电路设计	336
11.1.3	软件程序设计	297	11.5.3	软件程序设计	336
11.2	DS18B20 多点温度监测系统 设计	303	11.6	点阵 LED 显示屏设计	352
11.2.1	功能要求	303	11.6.1	功能要求	352
			11.6.2	硬件电路设计	352
			11.6.3	软件程序设计	353
			复习思考题 11		355
			附录 A	8051 指令表	356
			附录 B	Proteus 中的常用元器件	361
			参考文献		362

第 1 章 8051 单片机的基本组成

1.1 8051 单片机的特点与基本结构

8051 单片机是在美国 Intel 公司于 20 世纪 80 年代推出的 MCS-51 系列高性能 8 位单片机的基础上发展而来的,它在单一芯片内集成了并行 I/O 口、异步串行口、16 位定时器/计数器、中断系统、片内 RAM 和片内 ROM,以及其他一些功能部件。现在 8051 单片机已经有了很大的发展,除 Intel 公司之外,Philips、Siemens、AMD、Fujitsu、OKI、Atmel、SST、Winbond 等公司都推出了以 8051 为核心的新一代 8 位单片机,这种新型单片机的集成度更高,在片内集成了更多的功能部件,如 ADC、PWM、PCA、WDT 及高速 I/O 口等。不同公司推出的 8051 具有各自的功能特点,但它们的内核都是以 Intel 公司的 MCS-51 为基础的,并且指令系统兼容,从而给用户带来了更大的选择范围,同时又可以采用相同的开发工具。

8051 单片机在存储器的配置上采用的是所谓的“哈佛”结构,即在物理上具有独立的程序存储器和数据存储器,而在逻辑上则采用相同的地址空间,利用不同的指令和寻址方式进行访问,可分别寻址 64 KB 的程序存储器空间和 64 KB 的数据存储器空间,充分满足工业测量控制的需要。8051 单片机共有 111 条指令,其中包括乘、除指令和位操作指令。中断源有 5 个(8032/8052 为 6 个),分为两个优先级,每个中断源的优先级是可编程的。

在 8051 单片机的内部 RAM 区中开辟了 4 个通用工作寄存区,共有 32 个通用寄存器,可以适用于多种中断或子程序嵌套的情况。另外还在内部 RAM 中开辟了 1 个位寻址区,利用位操作指令可以对位寻址区中每个单元的每个位直接进行操作,特别适合于解决各种开关控制和逻辑问题。ROM 型 8051 在单芯片应用方式下其 4 个并行 I/O 口(P0~P3)都可以作为输入/输出之用,在扩展应用方式下则需要采用 P0 和 P2 口作为片外扩展地址总线之用。8051 单片机内部集成了两个(8032/8052 为 3 个)16 位定时器/计数器,可以十分方便地进行定时和计数操作,还集成了一个全双工的异步串行接口,可同时发送和接收数据,为单片机之间的相互通信或与上位机通信带来极大的方便。

8051 单片机的基本组成如图 1.1 所示,一个单片机芯片内包括:

- 中央处理器 CPU,它是单片机的核心,用于产生各种控制信号,并完成对数据的算术逻辑运算和传送;
- 内部数据存储器 RAM,用以存放可以读/写的数据;
- 内部程序存储器 ROM,用以存放程序指令或某些常数表格;
- 4 个 8 位的并行 I/O 口, P0、P1、P2 和 P3,每个口都可以用作输入或者输出;
- 两个(8051)或三个(8052)定时器/计数器,用作外部事件计数器,也可用作定时;
- 内部中断系统具有 5 个中断源,两个优先级的嵌套中断结构,可实现二级中断服务程序嵌套,每个中断源都可用软件程序规定为高优先级中断或低优先级中断;
- 一个串行接口,可用于异步接收/发送器;

- 内部时钟，但晶体和微调电容需要外接，振荡频率可以高达 40 MHz。
以上各部分通过内部总线相连。

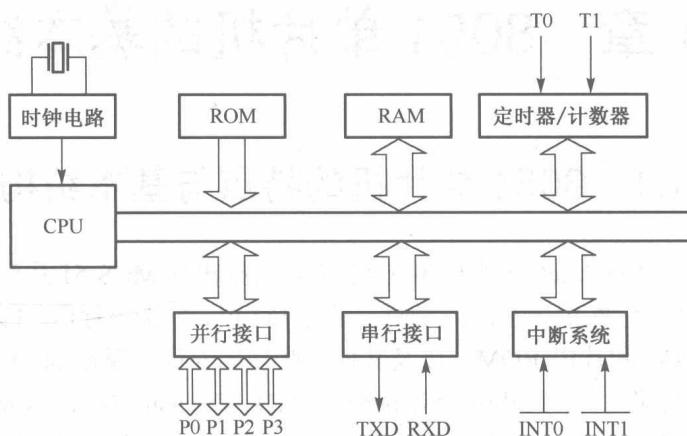


图 1.1 8051 单片机的基本组成

在很多情况下，单片机还要和外部设备或外部存储器相连，连接方式采用三总线（地址、数据、控制）方式，但在 8051 单片机中，没有单独的地址总线和数据总线，而是与通用并行 I/O 口中的 P0 口及 P2 口公用的，P0 口分时作为低 8 位地址线和 8 位数据线，P2 口则作为高 8 位地址线用，可形成 16 条地址线和 8 条数据线。

一定要建立一个明确的概念，即单片机在进行外部扩展时的地址线和数据线都不是独立的总线，而是与并行 I/O 口公用的，这是 8051 单片机结构上的一个特点。

图 1.2 所示为 8051 单片机的内部结构图，其中中央处理器 CPU 包含运算器和控制器两大部分，运算器完成各种算术和逻辑运算，控制器在单片机内部协调各功能部件之间的数据传送和运算操作，并对单片机外部发出若干控制信息。

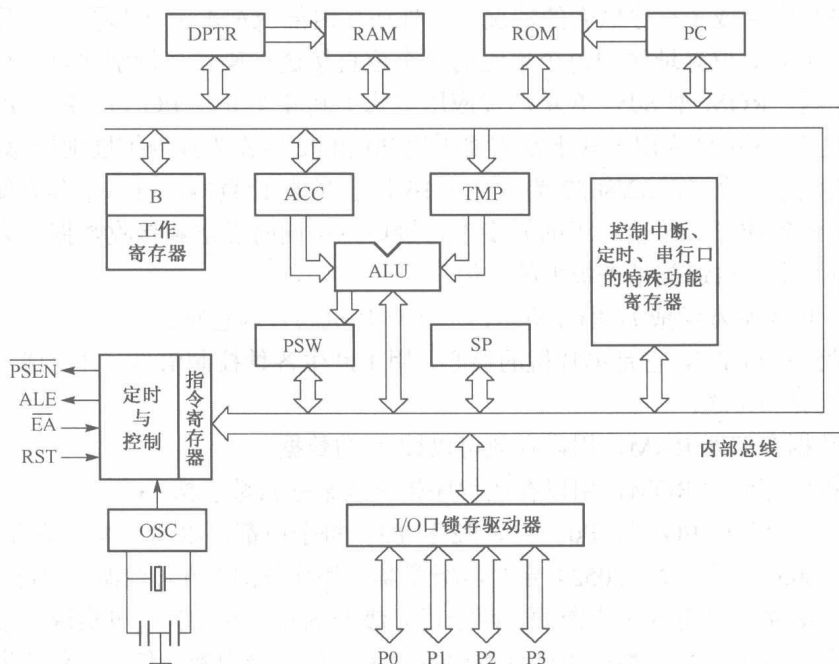


图 1.2 8051 单片机的内部结构

1. 运算器

运算器以算术逻辑单元 ALU 为核心，加上累加器 ACC、暂存寄存器 TMP 和程序状态字寄存器 PSW 等所组成。ALU 主要用于完成二进制数据的算术和逻辑运算，并通过对运算结果的判断影响程序状态字寄存器 PSW 中有关位的状态。累加器 ACC 是一个 8 位的寄存器(在指令中一般写为 A)，它通过暂存寄存器 TMP 与 ALU 相连，ACC 的工作最为繁忙，因为在进行算术逻辑运算时，ALU 的一个输入多为 ACC 的输出，而大多数运算结果也需要送到 ACC 中，在进行乘、除运算时，B 寄存器用来存放一个操作数，它也用来存放乘、除运算后的一部分结果，在不进行乘、除操作时，B 寄存器可用作通用寄存器。程序状态字寄存器 PSW 也是一个 8 位寄存器，用于存放运算结果的一些特征，格式如下：

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

其中 D7~D0 各标志位的意义如下。

CY：进位标志。在进行加法或减法运算时，若运算结果的最高位有进位或借位，CY=1，否则 CY=0，在执行位操作指令时，CY 作为位累加器。

AC：辅助进位标志。在进行加法或减法运算时，若低半字节向高半字节有进位或借位，AC=1，否则 AC=0，AC 还作为 BCD 码运算调整时的判别位。

F0：用户标志。用户可根据自己的需要对 F0 赋以一定的含义，如可以用软件来测试 F0 的状态以控制程序的流向。

RS1 和 RS0：工作寄存器组选择。可以用软件来置位或复位。它们与工作寄存器组的关系如表 1.1 所示。

表 1.1 RS1 和 RS0 与工作寄存器组的关系

RS1	RS0	工作寄存器组	片内 RAM 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：溢出标志。当两个带符号的单字节数进行运算，结果超出-128~+127 的范围时，OV=1，表示有溢出，否则 OV=0，表示无溢出。

D1：PSW 中的 D1 位为保留位，对于 8051 来说没有意义，对于 8052 来说为用户标志，与 F0 相同。

P：奇偶校验标志。每条指令执行完毕后，都按照累加器 A 中“1”的个数来决定 P 值，当“1”的个数为奇数时，P=1，否则 P=0。

2. 控制器

控制器包括定时控制逻辑、指令寄存器、指令译码器、程序计数器 PC、数据指针寄存器 DPTR、堆栈指针 SP、地址寄存器和地址缓冲器等。它的功能是对逐条指令进行译码，并通过定时和控制电路在规定的时刻发出各种操作所需的内部和外部控制信号，协调各部分的工作。下面简单介绍其中主要部件的功能。

程序计数器 PC: 用于存放下一条将要执行指令的地址。当一条指令按 PC 所指向的地址从程序存储器中取出之后, PC 的值会自动增加, 即指向下一条指令。

堆栈指针 SP: 用来指示堆栈的起始地址。8051 单片机的堆栈位于片内 RAM 中, 而且属于“上长型”堆栈, 复位后 SP 被初始化为 07H, 使得堆栈实际上由 08H 单元开始。必要时可以给 SP 装入其他值, 重新规定栈底的位置。堆栈中数据操作规则是“先进后出”, 每往堆栈中压入一个数据, SP 的值自动加 1, 随着数据的压入, SP 的值将越来越大, 当数据从堆栈弹出时, SP 的值将越来越小。

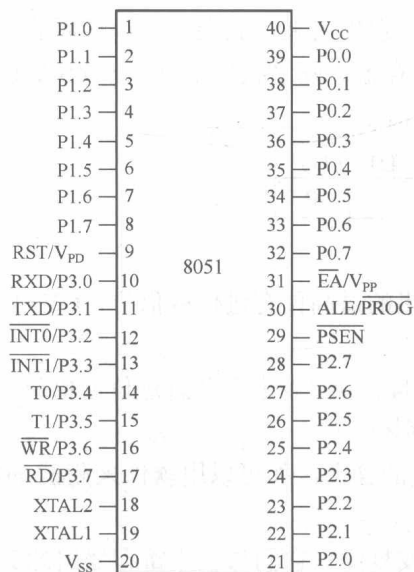


图 1.3 8051 单片机引脚分配图

指令译码器: 当指令送入指令译码器后, 由译码器对该指令进行译码, 即把指令转变成所需要的电平信号, CPU 根据译码器输出的电平信号使定时控制电路产生执行该指令所需要的各种控制信号。

数据指针寄存器 DPTR: 它是一个 16 位寄存器, 由高位字节 DPH 和低位字节 DPL 组成, 用来存放 16 位数据存储器的地址, 以便对片外 64 KB 的数据 RAM 区进行读/写操作。

采用 40 引脚双列直插封装 (PDIP) 的 8051 单片机引脚分配如图 1.3 所示。各引脚功能如下。

V_{SS} (20): 接地。

V_{CC} (40): 接+5 V 电源。

XTAL1 (19) 和 XTAL2 (18): 在使用单片机内部振荡电路时, 这两个端子用来外接石英晶体和微调电容, 如

图 1.4(a)所示。在使用外部时钟时, 则用来输入时钟脉冲, 但对 NMOS 和 CMOS 芯片接法不同, 图 1.4(b)所示为 NMOS 芯片 8051 外接时钟, 图 1.4(c)所示为 CMOS 芯片 80C51 外接时钟。

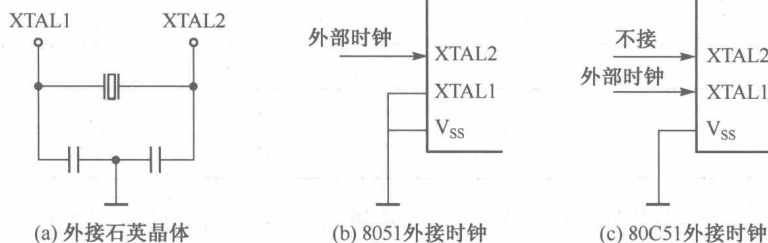


图 1.4 80C51 单片机的时钟接法

RST/ V_{PD} (9): RST 是复位信号输入端。当此输入端保持两个机器周期 (24 个振荡周期) 的高电平, 就可以完成复位操作。此引脚的第二功能是 V_{PD} , 即备用电源输入端, 当主电源发生故障, 降低到规定的低电平以下时, V_{PD} 将为片内 RAM 提供备用电源, 以保证存储在 RAM 中的信息不丢失。

ALE/ $\overline{\text{PROG}}$ (30): ALE 是地址锁存允许信号, 在访问外部存储器时, 用来锁存由 P0 口送出的低 8 位地址信号。在不访问外部存储器时, ALE 以振荡频率 1/6 的固定速率输出脉冲信号。因此它可用作对外输出的时钟。但要注意, 只要有外接存储器, 则 ALE 端输出的就不再是连续的周期脉冲信号了。第二功能 $\overline{\text{PROG}}$ 是用于对 8751 片内 EPROM 编程的脉冲输入端。

$\overline{\text{PSEN}}$ (29): 它是外部程序存储器 ROM 的读选通信号。在执行访问外部 ROM 指令的

时候,会自动产生 $\overline{\text{PSEN}}$ 信号,而在访问外部数据存储器 RAM 或访问内部 ROM 时,不产生 $\overline{\text{PSEN}}$ 信号。

$\overline{\text{EA}}/\text{V}_{\text{PP}}$ (31): 访问外部存储器的控制信号。当 $\overline{\text{EA}}$ 为高电平时,访问内部程序存储器,但当程序计数器 PC 的值超过 0FFFH (对于 8051/8051/8751 单片机) 或 1FFFH (对于 8052 单片机) 时,将自动转向执行外部程序存储器内的程序。当 $\overline{\text{EA}}$ 保持低电平时,则只访问外部程序存储器,而不管是否有内部程序存储器。该引脚的第二功能 V_{PP} 为对 8751 片内 EPROM 的 21V 编程电源输入。

P0.0~P0.7 (39~32): 双向 I/O 口 P0。P0 口首先作为双向 I/O 功能,其第二功能是在访问外部存储器时,可分时用作低 8 位地址和 8 位数据线,在对 8751 编程和校验时,用于数据的输入和输出。P0 口能以吸收电流的方式驱动 8 个 LS 型 TTL 负载。

P1.0~P1.7 (1~8): 双向 I/O 口 P1。P1 口能驱动 (吸收或输出电流) 4 个 LS 型 TTL 负载。在对 EPROM 编程和程序验证时,它接收低 8 位地址。在 8052 单片机中,P1.0 还用作定时器 2 的计数触发输入端 T2, P1.1 还用作定时器 2 的外部控制端 T2EX。

P2.0~P2.7 (21~28): 双向 I/O 口 P2。P2 口可以驱动 (吸收或输出电流) 4 个 LS 型 TTL 负载。其第二功能是在访问外部存储器时,输出高 8 位地址。在对 EPROM 编程和校验时,它接收高位地址。

P3.0~P3.7 (10~17): 双向 I/O 口 P3。P3 口能驱动 (吸收或输出电流) 4 个 LS 型 TTL 负载。P3 口的每条引脚都有各自的第二功能。

1.2 8051 单片机的存储器结构

图 1.5 所示为 8051 单片机的存储器结构图。在物理上它有 3 个存储器空间: 程序存储器 (CODE 空间)、片内数据存储器 (IDATA 和 DATA 空间)、片外数据存储器 (XDATA 空间)。访问不同存储器空间时须采用不同的指令。

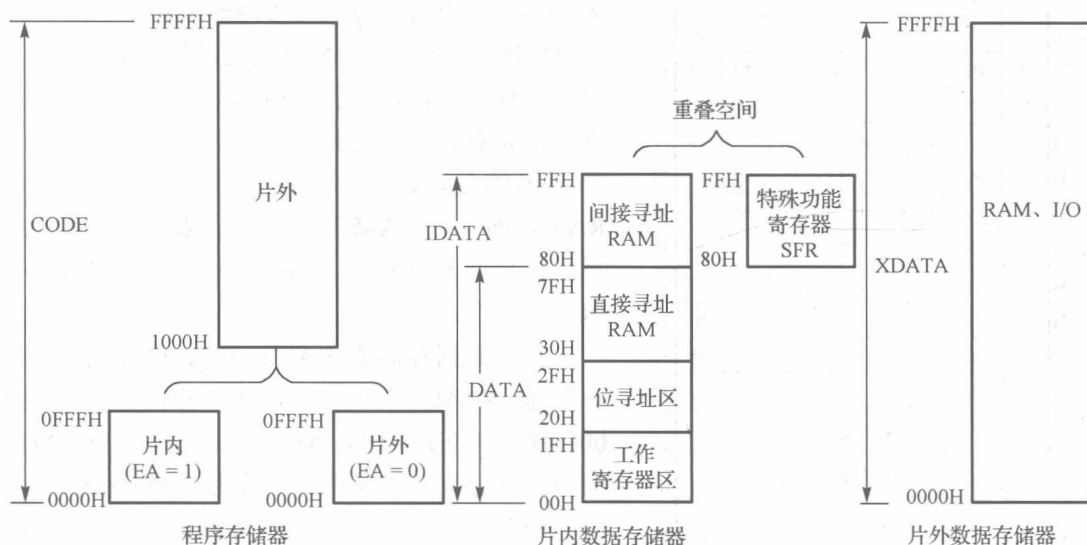


图 1.5 8051 单片机的存储器结构

程序存储器 ROM: 8051 单片机程序存储器 ROM 空间大小为 64KB, 地址范围为 0000H~

FFFFH, 用于存放程序代码和一些表格常数, 称为 CODE 空间。8051 单片机专门提供一个引脚 $\overline{\text{EA}}$ 来区分片内 ROM 和片外 ROM。 $\overline{\text{EA}}$ 引脚接高电平时, 单片机从片内 ROM 中读取指令, 当指令地址超过片内 ROM 空间范围后, 就自动地转向片外 ROM 读取指令; $\overline{\text{EA}}$ 引脚接低电平时, 所有的取指操作均对片外 ROM 进行。程序存储器的某些地址单元是保留给系统使用的: 0000H~0002H 单元是所有执行程序的入口地址, 复位后 CPU 总是从 0000H 地址开始执行程序; 0003H~002BH 单元均匀地分为 5 段, 用于 5 个中断服务程序的入口, 产生某个中断时, 将自动进入其对应入口地址开始执行中断服务程序, 一些新型 8051 单片机增加了更多的中断源, 它们的中断入口地址也相应增加。

片内数据存储器 RAM: 8051 单片机片内数据存储器 RAM 空间最大为 256 字节, 用于存放程序执行过程的各种变量及临时数据, 整个片内 RAM 地址范围 00H~FFH, 称为 IDATA 空间。片内 RAM 低 128 字节 (00H~7FH), 称为 DATA 空间, 它既可用直接寻址访问, 也可用间接寻址访问, 而片内 RAM 高 128 字节 (80H~FFH) 则只能采用间接寻址访问。片内 RAM 中 00H~1FH 地址范围称为工作寄存器区, 平均分为 4 组, 每组都有 8 个工作寄存器 R0~R7。在某一时刻, CPU 只能使用其中一组工作寄存器, 究竟选择哪一组工作寄存器, 则由程序状态字寄存器 PSW 中 RS0 和 RS1 的状态决定, 如表 1.1 所示。片内 RAM 中 20H~

2FH 地址范围称为位寻址区, 其中每个存储器单元的每一位称为一个 bit, 可以用位处理指令直接操作。

8051 单片机片内 RAM 位地址分配如图 1.6 所示。

8051 单片机在 IDATA 空间高 128 字节 (80H~FFH 地址范围) 安排了一个重叠空间, 称为特殊功能寄存器区 (又称 SFR 区), 地址也为 80~FFH, 但在使用时, 可通过指令加以区别。有些特殊功能寄存器是可以位寻址的, 其可寻址位称为 sbit。表 1.2 所示为 8051 单片机特殊功能寄存器地址及符号表, 表中带*的为可位寻址的特殊功能寄存器。片内 RAM 中的各个单元, 都可以通过其地址来表示, 而对于工作寄存器, 一般使用 R0~R7 表示, 对于特殊功能寄存器, 也是直接用其符号名较为方便。需要指出的是, 8051 单片机的堆栈默认使用片内 RAM, 而片内 RAM 空间十分有限, 因此要仔细安排堆栈指针 SP 的值, 以保证不会发生堆栈溢出, 导致系统崩溃。

片外数据存储器 RAM: 8051 单片机片外数据存储器 RAM 空间大小为 64 KB, 地址范围为 0000H~FFFFH, 称为 XDATA 空间。在 XDATA 空间内进行分页寻址操作时, 称为 PDATA 区。

8051 单片机采用所谓“哈佛”结构的存储器配置, 即在物理上具有独立的 ROM 存储器和片外 RAM 数据存储器, 而在逻辑上则采用相同的地址空间, 其地址范围都是 0000H~FFFFH, 单片机采用不

RAM 地址	MSB								LSB
7FH									127
2FH	7F	7E	7D	7C	7B	7A	79	78	47
2EH	77	76	75	74	73	72	71	70	46
2DH	6F	6E	6D	6C	6B	6A	69	68	45
2CH	67	66	65	64	63	62	61	60	44
2BH	5F	5E	5D	5C	5B	5A	59	58	43
2AH	57	56	55	54	53	52	51	50	42
29H	4F	4E	4D	4C	4B	4A	49	48	41
28H	47	46	45	44	43	42	41	40	40
27H	3F	3E	3D	3C	3B	3A	39	38	39
26H	37	36	35	34	33	32	31	30	38
25H	2F	2E	2D	2C	2B	2A	29	28	37
24H	27	26	25	24	23	22	21	20	36
23H	1F	1E	1D	1C	1B	1A	19	18	35
22H	17	16	15	14	13	12	11	10	34
21H	0F	0E	0D	0C	0B	0A	09	08	33
20H	07	06	05	04	03	02	01	00	32
1FH	工作寄存器3区								31
18H									24
17H									23
10H	工作寄存器2区								16
0FH									15
08H	工作寄存器1区								8
07H									7
00H	工作寄存器0区								0

图 1.6 8051 单片机片内 RAM 位地址分配

同的指令和寻址方式进行访问，同时通过不同的信号来选通 ROM 或片外 RAM。当从 ROM 中取指令时，采用信号 $\overline{\text{PSEN}}$ 来选通；而从外部 RAM 中读/写数据时，则采用 $\overline{\text{RD}}/\overline{\text{WR}}$ 信号来选通，从而可分别寻址 64 KB 的 ROM 程序存储器和 64 KB 的片外 RAM 数据存储器。

表 1.2 8051 单片机特殊功能寄存器地址及符号表

特殊功能寄存器符号	片内 RAM 地址	说 明
*ACC	E0H	累加器
*B	F0H	乘法寄存器
*PSW	D0H	程序状态字寄存器
SP	81H	堆栈指针
DPL	82H	数据指针（低 8 位）
DPH	83H	数据指针（高 8 位）
*IE	A8H	中断允许寄存器
*IP	B8H	中断优先级寄存器
*P0	80H	P0 口锁存器
*P1	90H	P1 口锁存器
*P2	A0H	P2 口锁存器
*P3	B0H	P3 口锁存器
PCON	87H	电源控制及波特率选择寄存器
*SCON	98H	串行口控制寄存器
SBUF	99H	串行数据缓冲器
*TCON	88H	定时器控制寄存器
TMOD	89H	定时器方式选择寄存器
TL0	8AH	定时器 0 低 8 位
TH0	8BH	定时器 0 高 8 位
TL1	8CH	定时器 1 低 8 位
TH1	8DH	定时器 1 高 8 位

1.3 CPU 时序

8051 单片机内部有一个高增益反相放大器，用于构成振荡器，反相放大器的输入端为 XTAL1，输出端为 XTAL2，分别是 8051 的第 19 和 18 脚。在 XTAL1 和 XTAL2 之间接一个石英晶体及两个电容，就可以构成稳定的自激振荡器，当振荡在 6~12 MHz 时通常取 30pF 左右的电容进行微调，如图 1.7 所示。晶体振荡器的振荡信号经过片内时钟发生器进行二分频，向 CPU 提供两相时钟信号 P1 和 P2。时钟信号的周期称为状态时间 S，它是振荡周期的两倍，在每个状态的前半周期 P1 信号有效，在每个状态的后半周期 P2 信号有效，CPU 就以这两相时钟信号为基本节拍指挥单片机各部分的协调工作。

CPU 执行一条指令所需要的时间是以机器周期为单位的，8051 单片机的一个机器周期包括 12 个振荡周期，分为 6 个 S 状态：S1~S6，每个状态又分为两拍，即

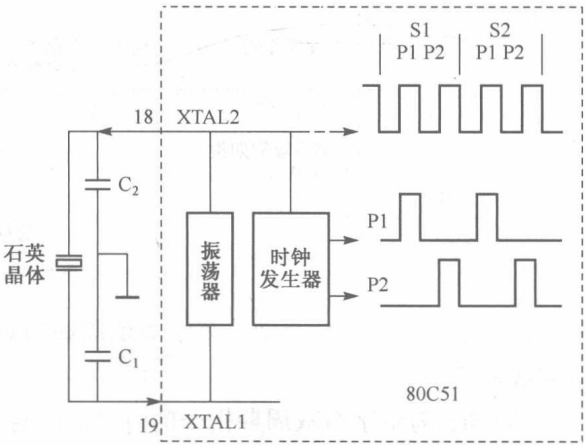


图 1.7 片内振荡器及时钟发生电路

前面介绍的 P1 和 P2 信号,因此一个机器周期中的 12 个振荡周期可表示为 S1P1, S1P2, S2P1, S2P2, ..., S6P1, S6P2。当采用 12 MHz 的晶体振荡器时,一个机器周期为 $1\mu\text{s}$ 。CPU 执行一条指令通常需要 1~4 个机器周期,指令的执行速度与其需要的机器周期数直接相关,所需机器周期数越少,速度越快。8051 单片机只有乘、除两条指令需要 4 个机器周期,其余均为单周期或双周期指令。

图 1.8 所示为几种典型的取指令和执行时序,从图中可以看到,在每个机器周期之内,地址锁存信号 ALE 两次有效,第一次出现在 S1P2 和 S2P1 期间,第二次出现在 S4P2 和 S5P1 期间。单周期指令的执行从 S1P2 开始,此时操作码被锁存在指令寄存器内。若是双字节指令,则在同一机器周期的 S4 状态读第 2 个字节。若是单字节指令,在 S4 状态仍进行读,但操作无效,且程序计数器 PC 的值不加 1。

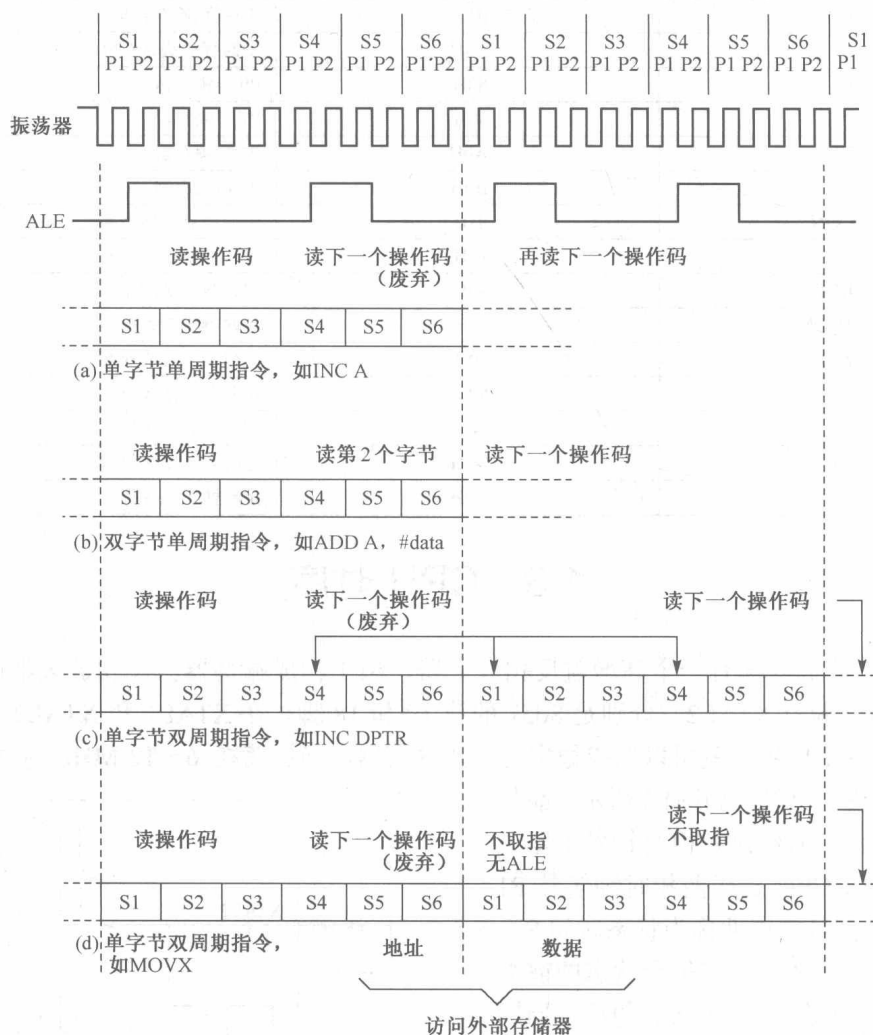


图 1.8 几种典型的取指令和执行时序

图 1.8(a)和图 1.8(b)分别为单字节单周期和双字节单周期指令的时序,它们都在 S6P2 结束时完成操作。

图 1.8(c)为单字节双周期指令的时序,在两个机器周期内进行 4 次操作,由于是单字节指令,所以后面的 3 次操作无效。

图 1.8(d)为 CPU 访问片外数据存储器指令“MOVX”的时序，它是一条单字节双周期指令，在第一个机器周期的 S5 状态开始送出片外数据存储器的地址，进行数据的读/写操作。在此期间没有 ALE 信号，所以在第二个周期不会产生取指操作。

1.4 复位信号与复位电路

8051 单片机与其他微处理器一样，在启动时需要复位，使 CPU 和系统的各个部件都处于一种确定的初始状态。复位信号从单片机的 RST 引脚输入，高电平有效，其有效电平应维持至少两个机器周期，若采用 6MHz 的晶体振荡器，则复位信号至少应持续 4μs 以上，才可以保证可靠复位。复位操作有上电自动复位和按键手动复位两种方式。上电自动复位是通过外部复位电路的电容充电来实现的，其电路如图 1.9(a)所示，只要电源 V_{CC} 电压上升时间不超过 1ms，通过在 V_{CC} 和 RST 之间加一个 22μF 的电容，RST 和 V_{SS} 引脚（即地）之间加一个 1kΩ 的电阻，就可以实现上电自动复位。按键手动复位电路如图 1.9(b)所示，它是在上电自动复位电路的基础上增加一个电阻 R1 和一个按键实现的。按下按键后，电容 C 通过 R1 放电，同时电源 V_{CC} 通过 R1 和 R2 分压，大部分电压都将落在 R2 上，从而使 RST 端得到一个高电平，导致单片机复位。

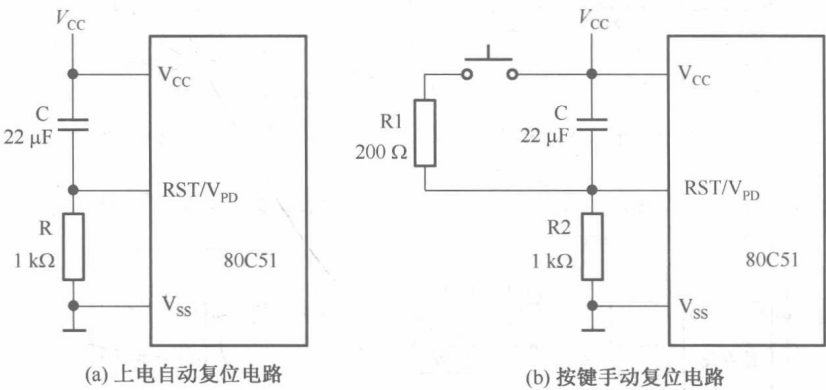


图 1.9 复位电路

当加上电源电压 V_{CC} 以后，RAM 的内容是随机的，复位不影响片内 RAM 的内容，复位后单片机内部各寄存器的状态如表 1.3 所示。

表 1.3 复位后单片机内部各寄存器的状态

寄存器	状 态	寄存器	状 态
PC	0000H	TMOD	00H
ACC	00H	TL0	00H
PSW	00H	TH0	000H
SP	07H	TL1	00H
DPTR	0000H	TH1	00H
P0~P3	FFH	SCON	00H
IP	××000000B	SBUF	不定
IE	0×000000B	PCON	0×××0000B