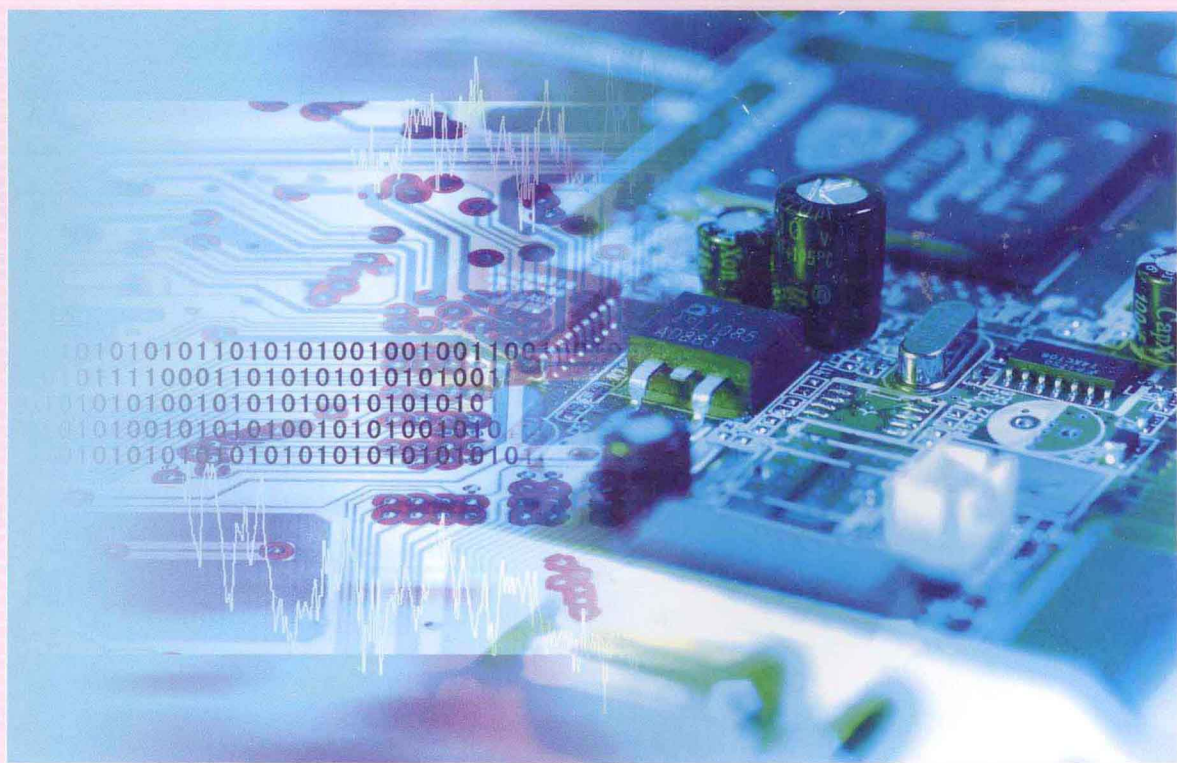




教育部高等职业教育示范专业规划教材

51系列单片机 项目实践

51XILIE DANPIANJI XIANGMU SHIJIAN



石长华 主编



赠电子课件等

 机械工业出版社
CHINA MACHINE PRESS



教育部高等职业教育示范专业规划教材

51 系列单片机项目实践

主 编 石长华
副主编 汪兆栋 凌 云
参 编 何 晖 欧阳晓慧
朱 斌 汪用瑜
主 审 刘焕成



机械工业出版社

本书以 51 系列单片机为对象,通过项目实践的方式,将单片机原理的主要内容有侧重地融入到 9 个项目实践中。各个项目的实现无需购置元器件和进行电路制作,使用电路设计与仿真软件 Proteus 和单片机程序集成开发软件 Keil μ Vision2,将单片机技术的硬件和软件、理论和实践、仿真实验和课程项目设计有机地结合起来,使学生在接近实际开发的过程中较完整地学习单片机原理及应用技术。本书项目中的所有实例均可在 Proteus 软件平台上直接运行。

本书在编程语言的教学上也进行了创新尝试,在第 2 章和第 3 章分别对单片机的汇编语言和 C 语言进行了介绍,除了“流水灯的设计”项目采用汇编语言编程,其余 8 个项目均采用 C 语言编程。

本书可作为高职高专电子信息类各专业的教学和实验用书,也可供学生进行课程设计、毕业设计和参加大学生电子设计竞赛时阅读参考。

为方便教学,本书配有免费电子课件、项目仿真电路和源程序等,凡选用本书作为授课教材的学校,均可来电或邮件索取,010-88379564 或 cmpqu@163.com。

图书在版编目(CIP)数据

51 系列单片机项目实践/石长华主编. —北京:机械工业出版社,2010.3
教育部高等职业教育示范专业规划教材
ISBN 978-7-111-29814-4

I. ①5… II. ①石… III. ①单片微型计算机—高等学校:技术学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2010)第 028652 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:曲世海 责任编辑:曲世海 李 宁

版式设计:霍永明 责任校对:纪 敬

封面设计:马精明 责任印制:李 妍

唐山丰电印务有限公司印刷

2010 年 5 月第 1 版第 1 次印刷

184mm x 260mm · 13.25 印张 · 323 千字

0001—4000 册

标准书号:ISBN 978-7-111-29814-4

定价:23.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

门户网:<http://www.cmpbook.com>

销售一部:(010) 68326294

教材网:<http://www.cmpedu.com>

销售二部:(010) 88379649

读者服务部:(010) 68993821 封面无防伪标均为盗版

前 言

单片机课程是计算机应用、通信、机械、自动化等专业学生的一门重要专业课，实践性环节较强，能将理论和实践相结合，更好地培养学生利用已学知识解决实际工程问题的能力。传统的单片机教学偏重于汇编指令和单片机资源介绍，程序实现一般都在实验室的实验箱上完成，程序和电路的改动极不方便，影响了学生学习兴趣和创新能力的培养。而利用 Proteus 软件实时仿真单片机工作，可以直接看到程序运行和电路工作过程与结果，弥补了理论教学、实验实践和工程应用脱节的缺陷。

本书主要有以下特点：

1) 采用项目实践方式，将单片机的原理教学融入到项目实验中，使学生在项目实验中学习单片机的原理。克服了传统单片机教材“原理书少实验例子，实验书少原理内容”的缺点。

2) 编程语言的教学上也进行了创新尝试，对汇编语言和 C 语言都进行了介绍，方便学生查找和对照学习，项目实验基本上用 C 语言进行程序设计。

3) 所有项目内容都用 Proteus 软件仿真实现，不受具体实验器材限制，边设计、边调试，在仿真的同时看到运行结果，从而节约购置元器件的费用，缩短了单片机开发应用的时间，激发学生学习兴趣。

4) DS18B20 数字温度计、电子时钟的设计、液晶显示器 www.cmpbook.com、LED 点阵显示汉字等项目实验具有实用性，可作为学生课外科技活动、电子设计竞赛、课程设计的参考用书。

5) 所有实验电路和源程序提供下载，方便学生课后研习，节省学生画图和输入程序的时间，学生可专注于电路原理和程序功能的理解。

由于软件本身的原因，本书中部分元器件符号采用的是非国标符号，具体对应关系请参考有关资料，特提请读者注意。

本书由石长华任主编，第 1、2 章由何晖、欧阳晓慧编写，第 3、5、6、7、9、10、11 章由汪兆栋、朱斌、汪用瑜编写，第 4、8 章由石长华编写，第 12、13 章由凌云编写。最后由石长华统稿并对各章进行了适当补充。刘焕成高级工程师对本书进行了审阅，提出了许多宝贵意见，在此表示感谢。

为方便教学，本书配有免费电子课件、项目仿真电路和源程序等，凡选用本书作为授课教材的学校，均可来电或邮件索取，010-88379564 或 cmpqu@163.com。

由于编者水平有限，错误和不足之处在所难免，敬请各位读者批评指正。

编 者

目 录

前言

第1章 单片机硬件系统基础 1

1.1 概述 1

1.1.1 微型计算机系统 1

1.1.2 单片机应用系统 1

1.2 MCS-51 系列单片机的基本结构 2

1.2.1 内部结构 2

1.2.2 信号引脚 3

1.2.3 时钟电路和复位电路 5

1.3 MCS-51 系列单片机的存储器结构 7

1.3.1 内部数据存储器 8

1.3.2 特殊功能寄存器 10

1.3.3 内部程序存储器 12

练习和思考题 12

第2章 单片机汇编语言基础 14

2.1 概述 14

2.2 MCS-51 系列单片机的指令格式 14

2.3 MCS-51 系列单片机的指令寻址方式 15

2.3.1 立即数寻址 15

2.3.2 直接寻址 15

2.3.3 寄存器寻址 15

2.3.4 寄存器间接寻址 16

2.3.5 变址寻址 16

2.3.6 相对寻址 17

2.3.7 位寻址 17

2.4 MCS-51 系列单片机的指令系统 17

2.4.1 指令分类 17

2.4.2 指令功能 18

2.4.3 指令执行的过程 18

2.5 MCS-51 系列单片机的数据传送指令 19

2.5.1 8 位数据传送指令 20

2.5.2 16 位数据传送指令 21

2.5.3 堆栈操作指令 21

2.5.4 字节交换指令 22

2.5.5 片外数据传送指令 22

练习和思考题 23

第3章 单片机 C 语言基础 26

3.1 数据结构 26

3.1.1 数据类型 26

3.1.2 常量与变量 27

3.1.3 数组 29

3.1.4 指针 32

3.2 运算符、表达式和优先级 34

3.2.1 赋值运算符 34

3.2.2 算术运算符及表达式 34

3.2.3 关系运算符、表达式及 优先级 35

3.2.4 逻辑运算符、表达式及 优先级 36

3.2.5 位操作运算符及表达式 36

3.2.6 复合赋值运算符及表达式 38

3.2.7 指针与地址运算符 39

3.3 C 语言程序基本结构与相关语句 39

3.3.1 C 语言程序的基本结构及 其流程图 39

3.3.2 选择语句 40

3.3.3 循环语句 42

3.3.4 循环的嵌套 44

3.3.5 break、continue 和 return 语句 44

3.4 函数 45

3.4.1 函数的分类 45

3.4.2 函数的定义 46

3.4.3 函数的参数和函数值 48

3.4.4 函数的调用与声明 49

练习和思考题 51

第4章 一步一步学做仿真实验	53
4.1 仿真实验：单片机控制单个LED 闪烁发光	53
4.2 仿真实验中用到的软件——Keil 快速入门	53
4.2.1 Keil 软件的界面	54
4.2.2 建立一个工程	55
4.2.3 产生一个 .hex 文件	58
4.3 仿真实验中用到的软件——Proteus 快速入门	61
4.3.1 Proteus 软件的界面	61
4.3.2 仿真单个LED 闪烁发光 电路	64
练习和思考题	70
第5章 并行输入/输出接口	72
5.1 项目内容：流水灯的设计	72
5.1.1 硬件电路设计	72
5.1.2 基本功能	73
5.2 程序设计	73
5.2.1 程序代码设计	73
5.2.2 仿真结果	75
5.3 相关知识	76
5.3.1 ORG 和 END 指令	76
5.3.2 逻辑移位指令	77
5.3.3 控制转移类指令	78
5.3.4 子程序调用与返回指令	80
5.3.5 并行输入/输出 P1 口的结构	81
5.3.6 LED 的应用	82
5.3.7 延时子程序的设计	83
5.4 知识综述：并行输入/输出接口	86
5.4.1 P0 口	86
5.4.2 P2 口	87
5.4.3 P3 口	87
练习和思考题	87
第6章 单片机驱动LED 数码管	89
6.1 项目内容：倒计时秒表设计	89
6.1.1 硬件电路设计	89
6.1.2 基本功能	89
6.2 程序设计	90
6.2.1 程序代码设计	90
6.2.2 仿真结果	91
6.3 相关知识	91
6.3.1 LED 数码管原理	91
6.3.2 LED 数码管字形码	93
6.4 知识扩展	94
6.4.1 LED 静态显示方式	94
6.4.2 LED 动态显示方式	95
练习和思考题	96
第7章 单片机的键盘控制	98
7.1 项目内容：简单计算器的设计	98
7.1.1 硬件电路设计	98
7.1.2 基本功能	99
7.2 程序设计	99
7.2.1 程序代码设计	99
7.2.2 仿真结果	104
7.3 相关知识：单片机系统中的键盘 接口技术	104
7.3.1 键盘工作原理及消抖	104
7.3.2 独立式键盘	106
7.3.3 矩阵式键盘	108
7.4 8255A 并行输入/输出接口扩充 键盘	111
7.4.1 8255A 的基础知识	111
7.4.2 8255A 键盘的扩充	114
练习和思考题	115
第8章 单片机的中断系统	116
8.1 项目内容：八路抢答器	116
8.1.1 基本功能	116
8.1.2 硬件电路设计	116
8.2 程序设计	117
8.3 相关知识	120
8.3.1 中断的基本概念	120
8.3.2 MCS-51 系列单片机的中断 系统	121
8.3.3 与中断有关的寄存器	121
8.3.4 汇编语言中断程序设计要点	124
8.3.5 C 语言中断程序设计要点	125
8.3.6 中断应用举例	126

练习和思考题	129	11.3 相关知识	160
第9章 定时/计数器	130	11.3.1 A/D 转换器概述	160
9.1 项目内容: 电子时钟的设计	130	11.3.2 ADC0809 的内部结构和 信号引脚	162
9.1.1 硬件电路设计	130	11.3.3 8051 单片机与 ADC0809 的 接口	164
9.1.2 基本功能	131	11.3.4 D/A 转换器综述	165
9.2 程序设计	131	11.3.5 8051 单片机与 DAC0832 的 接口	167
9.2.1 程序代码设计	131	练习和思考题	170
9.2.2 仿真结果	134	第12章 单片机与液晶显示器	171
9.3 相关知识: 定时/计数器综述	135	12.1 项目内容: 液晶显示器显示 www.cmpbook.com	171
9.3.1 定时/计数器的结构和工作 原理	135	12.1.1 硬件电路设计	171
9.3.2 定时/计数器的工作方式寄存器 和控制寄存器	136	12.1.2 基本功能	172
9.3.3 定时/计数器的工作方式	138	12.2 程序设计	172
9.3.4 定时/计数器的初始化	139	12.2.1 程序代码设计	172
9.3.5 定时/计数器的应用举例	140	12.2.2 仿真结果	175
练习和思考题	144	12.3 相关知识	175
第10章 单片机串行接口通信	145	12.3.1 LCM1602 外观及功能	175
10.1 项目内容: LED 点阵显示汉字	145	12.3.2 LCM1602 与单片机连接	176
10.1.1 硬件电路设计	145	12.3.3 LCM1602 的读写操作	177
10.1.2 基本功能	145	12.3.4 LCM1602 的指令	179
10.2 程序设计	146	12.3.5 LCM1602 字符显示位的 控制	181
10.2.1 程序代码设计	146	12.3.6 LCM1602 的初始化	181
10.2.2 仿真结果	150	12.3.7 LCM1602 的自定义字符	182
10.3 相关知识	150	练习和思考题	182
10.3.1 字符和汉字显示的基本原理	150	第13章 测温元件 DS18B20 应用	183
10.3.2 字符、汉字模块的生成	152	13.1 项目内容: DS18B20 数字 温度计	183
10.3.3 LED 点阵模块	152	13.1.1 硬件电路设计	183
10.3.4 LED 点阵的驱动方法	153	13.1.2 基本功能	184
10.3.5 单片机的串行接口工作 模式 0	153	13.2 程序设计	184
练习和思考题	155	13.2.1 程序代码设计	184
第11章 单片机与 A/D 接口	156	13.2.2 仿真结果	191
11.1 项目内容: 数字电压表的设计	156	13.3 相关知识: DS18B20 介绍和 1-Wire 总线接口	191
11.1.1 硬件电路设计	156	13.3.1 DS18B20 的外观及功能	191
11.1.2 基本功能	157		
11.2 程序设计	157		
11.2.1 程序代码设计	157		
11.2.2 仿真结果	160		

13.3.2 DS18B20 的存储器结构	192	附录 A Proteus 常用元器件英文	
13.3.3 DS18B20 的命令序列及		缩写表	196
操作时序	193	附录 B MCS-51 指令表	197
练习和思考题	195	参考文献	202
附录	196		

第 1 章 单片机硬件系统基础

1.1 概述

从计算机的发明到现在已经有半个多世纪的历程，现代科学技术对计算机的依赖和超大规模集成电路与微处理器技术的长足进步，极大地推动了计算机产品的发展。计算机已经普遍应用在不同的领域和场所，成为现代社会必不可少的信息处理工具。其中，单片机的应用更是广泛。

1.1.1 微型计算机系统

微处理器是在 20 世纪 70 年代研制出来的。在一片大规模的集成电路上集成算术逻辑运算部件、控制部件、内部总线、通用寄存器、指令与数据储存器等，以微处理器为中央处理器并配置一定的存储器和输入/输出接口而组成的计算机称为微型计算机。随着半导体集成电路技术的不断进步，微处理器芯片的集成速度快速增长，功能越来越强大，微型计算机的性能随之得到提高，应用场合不断扩大，品种繁多，如个人计算机(台式机、便携机)、手持式计算机。微型计算机是近年来计算机应用领域中发展最快的一个分支。

微型计算机系统由硬件和软件两大部分组成。硬件系统是指构成计算机系统的实体和装置，通常由运算器、控制器、存储器、输入/输出设备组成。其中，运算器和控制器一般制作在一块集成芯片上，称为中央处理单元(CPU)，又称为中央处理器，它是微型计算机的核心部件。软件系统是微型计算机系统所用的各种程序的总称。软件系统与硬件系统共同构成完整的微型计算机系统，两者相辅相成，缺一不可。

1.1.2 单片机应用系统

1. 单片机的概念

在同一芯片上不仅集成了微处理器电路，还集成了大容量存储器、输入/输出控制等接口电路，这样的大规模和超大规模集成电路称为单片机计算机(简称单片机)或嵌入式计算机。这种计算机可以作为一个信息处理部件，嵌入到应用系统和设备中，执行特定数据处理和控制的任务。对于嵌入式计算机，用户不能像使用普通计算机那样直接对计算机操作，而是对设备进行操作，计算机则按事先编制的程序，根据设备或系统的要求完成相应的数据处理和控制的功能。例如，手机、MP3、商务通、数码相机、打印机等产品中都含有嵌入式计算机并将其作为主要部件。计算机的嵌入式应用已成为计算机应用的主要形式之一。

2. 单片机应用系统的组成

目前所称的单片机计算机属于嵌入式计算机，结构和功能相对简单，即集成在一块芯片上的微型计算机。把组成微型计算机的各种功能部件，包括 CPU、随机存储器(RAM)、只读存储器(ROM)、基本输入/输出(I/O)接口电路、定时/计数器、中断系统等部分制作在一

块集成芯片上，构成一个完整的微型计算机，从而实现微型计算机的基本功能。

单片机应用系统是以单片机为核心，配以输入、输出、显示、控制等外围电路和软件，能实现一种或多种功能的应用系统。

单片机应用系统的设计人员必须从硬件和软件两个角度来深入了解单片机技术，并将两者有机结合起来，才能形成具有特定功能的应用系统和整机产品。

3. MCS-51 系列单片机

单片机的种类很多，其中 MCS-51 系列单片机结构典型，应用广泛。本书以 MCS-51 系列单片机为对象，以 9 个应用项目为载体，逐个介绍单片机的硬件结构、工作原理、汇编语言和 C 语言及应用系统的设计。

MCS-51 系列单片机有各种产品型号，可分为 51 系列和 52 系列。表 1-1 中列出了几种型号的主要性能指标。

表 1-1 MCS-51 系列单片机的主要性能指标

子系列	片内 ROM 形式				片内存储容量		寻址范围	I/O 特性		
	无	ROM	EP ROM	Flash ROM	ROM	RAM		计数器	并行接口	串行接口
51	8031	8051	8751	8951	4KB	128B	64KB	2 个	4 个	1 个
	80C31	80C51	87C51	89C51	4KB	128B	64KB	16 位	8 位	全双工
52	8032	8052	8752	8952	8KB	256B	64KB	3 个	4 个	1 个
	80C32	80C52	87C52	89C52	8KB	256B	64KB	16 位	8 位	全双工

1.2 MCS-51 系列单片机的基本结构

1.2.1 内部结构

MCS-51 系列单片机由 CPU、内部数据存储器(RAM)、内部程序存储器(ROM)、定时/计数器、并行接口、串行接口、中断系统和时钟电路组成。内部结构示意图如图 1-1 所示。

1. CPU

CPU 由运算器和控制器组成，它是单片机的控制中心，负责各个部件协调一致地工作。

2. RAM

RAM 用于存放计算和控制过程中的数据，单元内的数据可读写，掉电后信息会丢失。RAM 可分为系统使用单元和用户单元，如累加器(A)、工作寄存器(R0 ~ R7)和特殊功能寄存器等属于系统使用单元；用户区的存储单元可存放用户输入和输出的数据。

3. ROM

ROM 用于存放控制单片机工作的程序，掉电后程序不会丢失。单片机工作之前必须先将编制好的应用程序下载至芯片的 ROM 中。

4. 定时/计数器

实现对内部时钟和外部信号的计数功能。当设定的定时/计数数值满足一定的条件后，

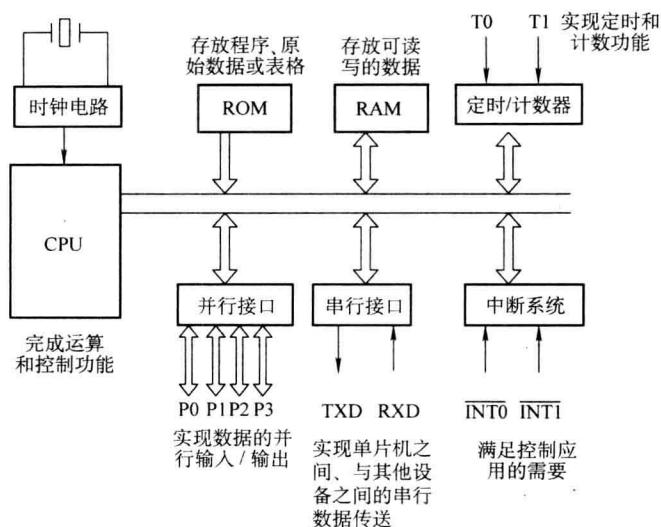


图 1-1 MCS-51 系列单片机的内部结构示意图

定时/计数器会做出标记通知 CPU，CPU 响应后完成相应操作。

5. 并行接口

并行接口提供与外部输入/输出设备的连接，可以按 8 位并行方式使用，也可一位一位使用。

6. 串行接口

串行接口提供与外部串行输入/输出设备的连接或通信，只能一位一位地使用。

7. 中断系统

中断系统提高了单片机对外部意外事件的感知能力。当外部某一事件发生时，CPU 能及时知道、响应并进行相应的处理。

8. 时钟电路

时钟电路为单片机各部件的工作提供统一的时钟，使各部件能在统一的节拍下进行协调一致的工作。

图 1-2 为 ATMEL 公司 AT89S51 的内部结构图。

1.2.2 信号引脚

图 1-3 是 MCS-51 系列单片机 40 引脚双列直插式集成电路芯片引脚图。

1. 输入/输出接口信号引脚

P0.0 ~ P0.7：P0 口，8 位双向口。

P1.0 ~ P1.7：P1 口，8 位双向口。

P2.0 ~ P2.7：P2 口，8 位双向口。

P3.0 ~ P3.7：P3 口，8 位双向口。

2. 控制信号引脚

ALE：地址锁存控制信号。ALE 以晶振固定频率的 1/6 输出正脉冲。在系统扩展时，ALE 用于把 P0 口输出的低 8 位地址锁存，以实现低 8 位地址和数据的隔离。

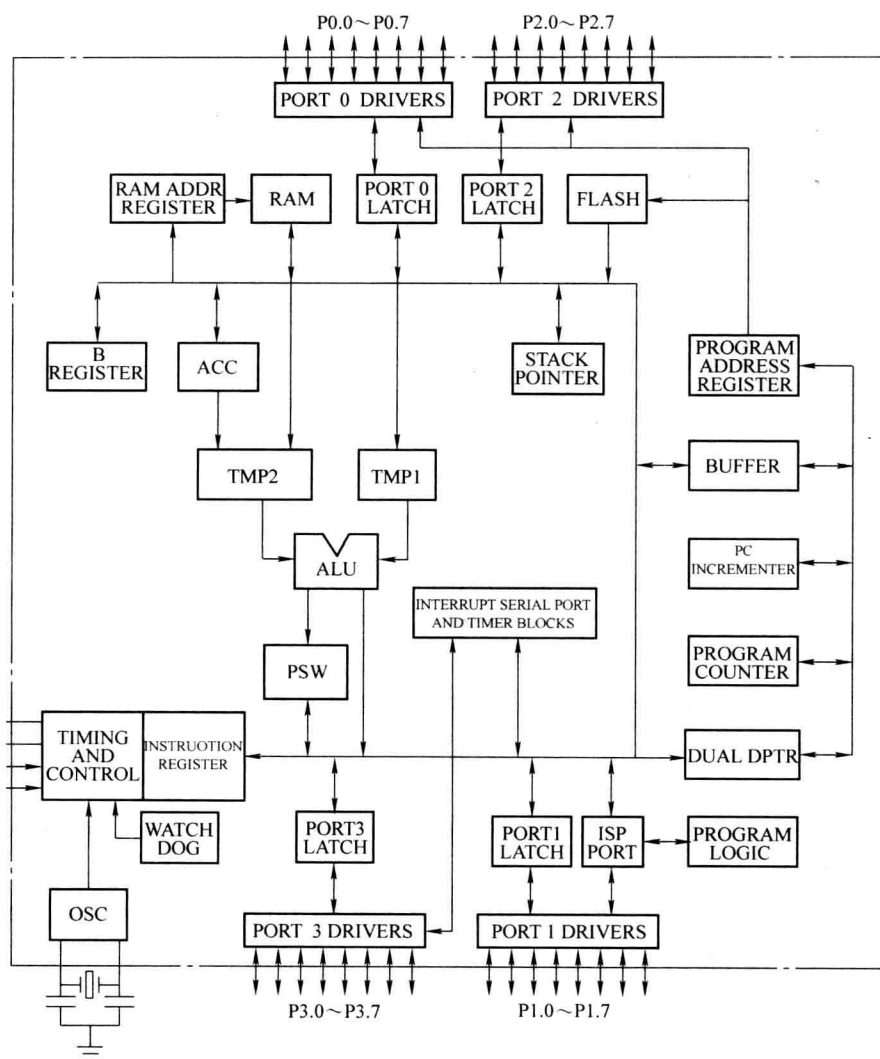


图 1-2 AT89S51 的内部结构图

$\overline{\text{PSEN}}$: 外部程序存储器的读选通信号。在读外部 ROM 时, $\overline{\text{PSEN}}$ 有效(低电平), 以实现外部 ROM 单元的读写操作。

$\overline{\text{EA}}$: 访问程序存储器控制信号。当 $\overline{\text{EA}}$ 信号为低电平时, 对 ROM 的读操作限定在外部程序存储器; 当 $\overline{\text{EA}}$ 信号为高电平时, 对 ROM 的操作从内部程序存储器开始, 延至外部程序存储器。

RST: 复位信号。当该引脚的信号为高电平, 并延续两个机器周期以上时, 完成单片机的复位, 内部相应单元完成初始化, 单片机进入工作状态。当单片机正常工作时, RST 为低电平。

3. 其他信号引脚

XTAL1 和 XTAL2: 外接晶体引线端。当使用芯片内部时钟时, 用于外接石英晶体和微调电容。当使用外部时钟时, 用于接外部时钟脉冲信号。

V_{SS} : 地线。

V_{CC} : +5V 电源。

4. P3 口第二功能信号引脚

P3.0(RXD): (串行数据接收)。

P3.1(TXD): (串行数据发送)。

P3.2($\overline{INT0}$): (外部中断0 申请)。

P3.3($\overline{INT1}$): (外部中断1 申请)。

P3.4(T0): (定时/计数器0 外部输入)。

P3.5(T1): (定时/计数器1 外部输入)。

P3.6($\overline{WR}$): (外部数据存储器写脉冲)。

P3.7($\overline{RD}$): (外部数据存储器读脉冲)。

5. RPPROM 程序存储器固化

编程脉冲: ALE/PROG。

编程电压(25V): $\overline{EA}/V_{PP}$ 。

备用电源引入: RST/ V_{PD} 。

另外, 对于 AT89S51 芯片, 内部包含 Flash ROM,

P1.5、P1.6 和 P1.7 用于程序固化(下载)使用, 与内部 EPROM 的芯片下载不同。

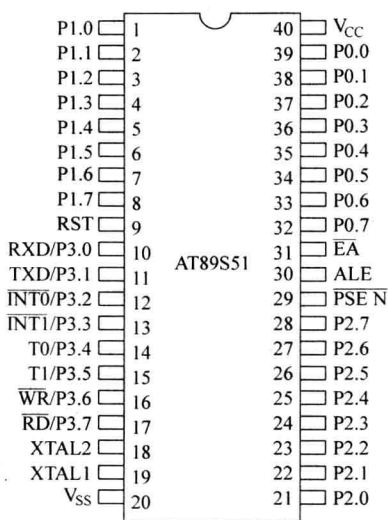


图 1-3 MCS-51 系列单片机 40 引脚

双列直插式集成电路芯片引脚图

1.2.3 时钟电路和复位电路

单片机要正常工作, 必须有时钟电路和复位电路。单片机的最小运行环境由单片机芯片、复位电路和时钟电路组成。

1. 时钟电路

时钟电路用于产生单片机工作所需要的时钟信号。时钟是计算机的“心脏”, 它控制着单片机的工作节奏。

(1) 内部时钟电路 当使用内部时钟电路时, 外接晶振频率范围为 1.2 ~ 12MHz。外接晶振频率越高, 系统的工作频率也就越高, 一般晶振频率选择 12MHz。图 1-4 是外部晶振的典型接法, 参数值可根据实际需要调整。 C_1 和 C_2 为微调电容, 一般选 30pF 左右。

(2) 使用外部时钟电路 在单片机系统中, 如果外部已经有时钟电路, 而且单片机的时钟需要与它一致时, 或在单片机系统需要单片机信号同步时, 则一般选择外部时钟方式。图 1-5 是外部时钟信号的一般连接图。

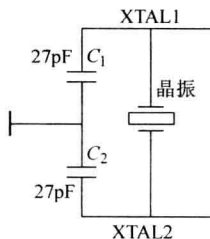


图 1-4 外部晶振的连接

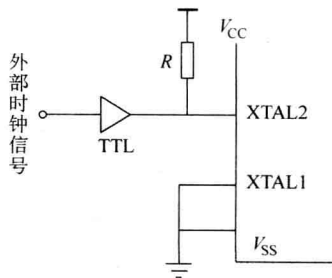


图 1-5 外部时钟信号的连接

(3) 周期的概念 振荡电路产生的振荡脉冲并不直接使用,而是经分频后再被系统所用。振荡脉冲经二分频后才作为系统的时钟信号。在二分频的基础上再三分频产生 ALE 信号,在二分频的基础上再六分频得到机器周期信号。振荡电路的内部结构如图 1-6 所示。

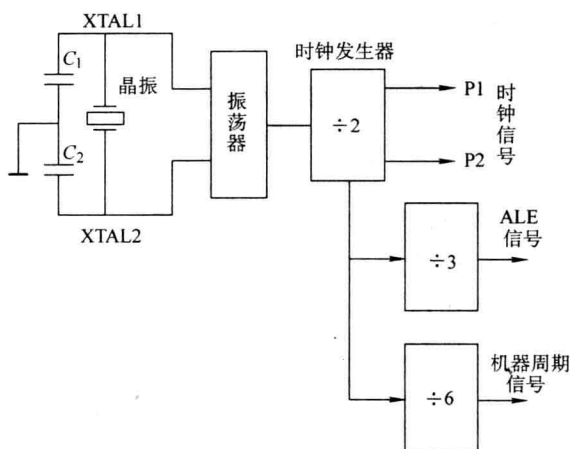


图 1-6 振荡电路的内部结构

1) 振荡周期(又称为节拍,用 P 表示): 外接晶振的周期。

2) 时钟周期(又称为状态,用 S 表示): 外部振荡脉冲经过二分频后的周期。一个状态包含两个节拍,前一个称为 P1,后一个称为 P2。

3) 机器周期(T): 一个机器周期为 6 个状态,12 个节拍($1T=6S=12P$)。

一个机器周期是指 CPU 访问存储器一次所需要的时间,规定一个机器周期的宽度为 6 个状态,并依次记为 S1~S6,一个状态又包括两个节拍 P1 和 P2,因此一个机器周期总共有 12 个节拍,记为 S1P1、S1P2、…、S6P1、S6P2。

振荡周期、时钟周期与机器周期的关系如图 1-7 所示。

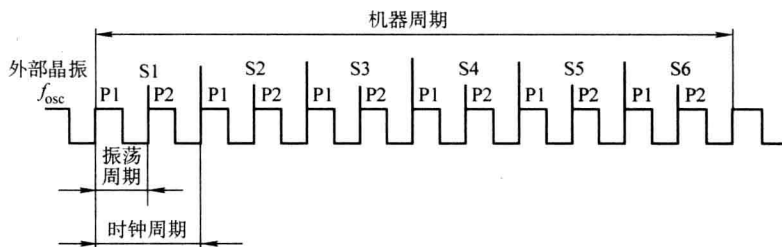


图 1-7 振荡周期、时钟周期与机器周期的关系

例如,假设采用外部晶振频率 $f_{osc}=6\text{MHz}$,则

$$\text{振荡周期} = 1/f_{osc} = 1/6 \times 10^{-6}\text{s}$$

$$\text{机器周期} = 12 \times \text{振荡周期} = 12 \times 1/6 \times 10^{-6}\text{s} = 2\mu\text{s}$$

(4) 指令周期 执行一条指令所需要的时间称为指令周期。不同指令的指令周期不同,有单机器周期指令、双机器周期指令和四机器周期指令。

2. 复位电路

当单片机复位时,CPU 和系统中的其他功能部件都处在一个确定的初始状态,并从这个状态开始工作,如程序计数器 $\text{PC}=0000\text{H}$,则从第一个单元取指令执行。

复位的条件是:必须使 RST 引脚加上持续两个机器周期以上的高电平,单片机才能有效复位。单片机的复位电路如图 1-8 所示。

单片机复位后,其片内各寄存器的初始状态见表 1-2。

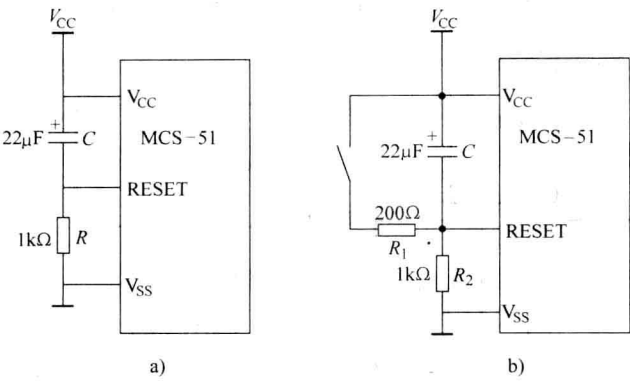


图 1-8 复位电路

a) 上电复位电路 b) 上电与按键复位电路

表 1-2 单片机复位后片内各寄存器的初始状态

寄存器名称	复位后的值	寄存器名称	复位后的值
程序计数器(PC)	0000H	定时/计数器方式寄存器(TMOD)	00H
累加器(A)	00H	定时/计数器控制寄存器(TCON)	00H
通用寄存器(B)	00H	定时器0 高字节 TH0	00H
程序状态寄存器(PSW)	00H	定时器0 低字节 TL0	00H
堆栈指针(SP)	07H	定时器1 高字节 TH1	00H
地址寄存器(DPTR)	0000H	定时器1 低字节 TL1	00H
并行接口锁存器(P0 ~ P3)	FFH	串行接口控制寄存器(SCON)	00H
中断优先级控制寄存器(IP)	*** 00000B	串行接口数据缓冲器(SBUF)	不定
中断允许控制寄存器(IE)	0 ** 00000B	电源及波特率寄存器(PCON)	0 ** 00000B

1.3 MCS-51 系列单片机的存储器结构

计算机的存储空间一般分为存放程序和存放数据两类，存储配置有两种典型结构：哈佛结构和普林斯顿结构。

- (1) 哈佛结构 程序空间(ROM)和数据空间(RAM)分为两个队列寻址。
- (2) 普林斯顿结构 程序空间(ROM)和数据空间(RAM)同在一个空间队列寻址。

本书介绍的 MCS-51 系列单片机采用的是哈佛结构的存储结构。MCS-51 系列单片机的存储器分为 ROM 和 RAM 两类。MCS-51 系列单片机内部有 4KB 的 ROM，最大可扩展到 64KB，所以可使用的最大程序空间为 64KB；内部有 128B 的 RAM，还可以扩展外部 RAM 共 64KB。访问外部设备与访问 RAM 一样，外部设备是与 RAM 统一编址的。MCS-51 系列单片机可访问的片外 RAM 和外设单元共 64KB。MCS-51 系列单片机对 ROM 的访问和片内 RAM 的访问是用不同指令实现的。

从物理空间看，单片机的存储器结构较为复杂，分为 4 个部分，即片内 ROM、片外

ROM、片内 RAM 和片外 RAM，如图 1-9 所示。但从逻辑空间上看，实际上存在 3 个独立的

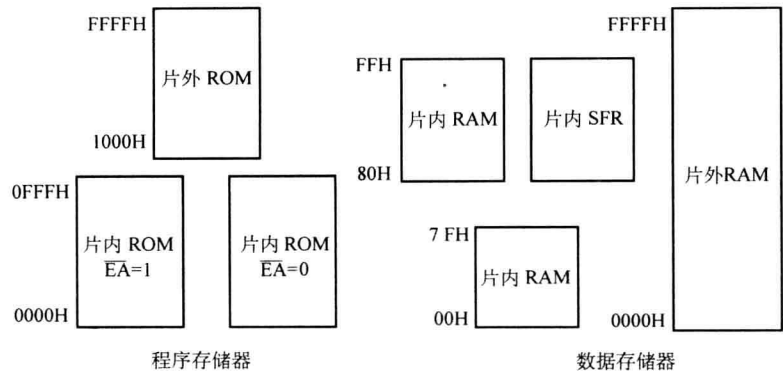


图 1-9 单片机的存储器结构

空间。片内、片外的程序存储器在同一个逻辑空间，它们的地址为 0000H ~ FFFFH (64KB)，是连续的；片内的数据存储器占一个逻辑空间，地址为 00H ~ FFH (256B)；片外的数据存储器占一个逻辑空间，地址为 0000H ~ FFFFH (64KB)。MCS-51 系列单片机会用不同的指令去访问不同的存储器空间。

1.3.1 内部数据存储器

MCS-51 系列单片机的内部数据存储器 (RAM) 共有 256 个单元，通常把 256 个单元分成两部分：低 128 个单元 (00H ~ 7FH) 和高 128 个单元 (80H ~ FFH)，如图 1-10 所示。

1. 低 128 个单元内部数据存储器存储区

低 128 个单元内部数据存储器按其用途分为寄存器区、位寻址区和用户数据区 3 个区域 (见表 1-3)。

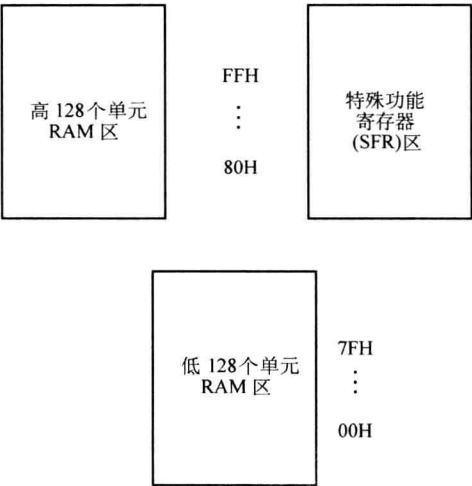


图 1-10 内部数据存储器

表 1-3 内部数据存储器的地址分配表

功 能 区 域	字 节 地 址	片内 RAM 存储区域(低 128 个单元)
用户数据区	30H ~ 7FH	数据存储器
位寻址区	20H ~ 2FH	可位寻址区
寄存器区	18H ~ 1FH	工作寄存器 3 组 (R7 ~ R0)
	10H ~ 17FH	工作寄存器 2 组 (R7 ~ R0)
	08H ~ 0FH	工作寄存器 1 组 (R7 ~ R0)
	00H ~ 07H	工作寄存器 0 组 (R7 ~ R0)

(1) 寄存器区 MCS-51 系列单片机共有 4 组工作寄存器，每组 8 个单元，用 R0 ~ R7

编号。它们用来暂存数据及中间结果，使用灵活。4 组工作寄存器占内部 RAM 的 00H~1FH 单元。在某一时刻，CPU 只能使用其中的一组工作寄存器(称为当前寄存器)。它由程序状态寄存器(PSW)中的 RS1、RS0 位的状态决定。单片机复位后，RS1 和 RS0 默认为工作寄存器 0 组。

(2) 位寻址区 内部 RAM 的 20H~2FH 单元既可以字节寻址，作为一般的 RAM 单元使用，又可以位寻址，进行布尔操作。具体的位地址见表 1-4。

表 1-4 内部 RAM 的 20H~2FH 单元的位地址

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

(3) 用户数据区 内部 RAM 的 30H~7FH 单元是提供用户使用的数据区。用户的数据存放在此区域，在实际使用时，常把堆栈开辟在此。

2. 高 128 个单元特殊功能寄存器(SFR)区

对于 MCS-51 系列单片机，在内部数据存储器 80H~FFH 的高 128 个单元中，特殊功能寄存器只占用其中的 21 个单元，其余单元无定义，用户不能对这些单元进行读写操作。对于 MCS-51 系列单片机，高 128 个单元除供专用寄存器(26 个)使用外，剩余地址单元可作为一般数据单元使用。MCS-51 系列单片机可寻址的特殊功能寄存器见表 1-5。

表 1-5 特殊功能寄存器

标 识 符	名 称	地 址	可否位寻址
PO	PO 口锁存器	80H	√
SP	堆栈指针	81H	×
DPTR	数据指针(DPH、DPL)	83H、82H	×
PCON	电源控制	87H	×