

高等职业教育机电系列精品教材

单片机 应用技术

DANPIANJI YINGYONG JISHU

主 编 马建民

副主编 张松枝 刘煜辉

王宏淼



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书共分为七个项目,分别为单片机最小系统设计、简易信号灯设计、定时器/计数器及中断系统应用、显示器与键盘接口技术应用、串行接口技术、D/A 与 A/D 转换接口技术、单片机系统扩展设计。

本书根据编者教学经验,在编写过程中遵循“项目化教学和任务驱动教学”原则,以应用为目的,以具体的任务为载体,将单片机的主要知识点分解到难度逐渐增加的项目任务中,通过做学结合让学生轻松学习单片机的理论和实践知识,掌握单片机的技术及其应用技能。

本书既可作为高等职业教育电气自动化技术及机电一体化技术等专业的教材,也可供相关企业技术人员参考。

图书在版编目(CIP)数据

单片机应用技术 / 马建民主编. — 上海 : 上海交通大学出版社, 2023

ISBN 978-7-313-30032-4

I. ①单… II. ①马… III. ①微控制器—高等职业教育—教材 IV. ①TP368.1

中国国家版本馆 CIP 数据核字 (2023) 第 257547 号

单片机应用技术

DANPIANJI YINGYONG JISHU

主 编: 马建民

出版发行: 上海交通大学出版社

邮政编码: 200030

印 制: 三河市骏杰印刷有限公司

开 本: 787 mm×1 092 mm 1/16

字 数: 222 千字

版 次: 2023 年 12 月第 1 版

书 号: ISBN 978-7-313-30032-4

定 价: 38.00 元

地 址: 上海市番禺路 951 号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 10.75

印 次: 2023 年 12 月第 1 次印刷

版权所有 侵权必究

告读者: 如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话: 0316-3662258

Preface

前言

职业教育的发展必须与产业的发展需求紧密结合。随着新技术的不断涌现和产业结构的调整升级,职业教育必须及时调整和更新课程设置,加强与企业合作,提高学生的职业技能和就业能力。“单片机应用技术”是高等职业院校电气自动化技术、机电一体化技术等专业学生学习和掌握嵌入式应用系统的入门课程。

本书主要讲授 51 单片机的系统结构、编程思想、时钟系统、中断技术、定时器/计数器、D/A 与 A/D 转换、扩展模块等内容。本书配有丰富的数字资源,采用项目化教学方式,每个项目对应典型任务,每个任务又配有源程序代码、Proteus 仿真电路图、仿真动画二维码、PPT 等资源。本书将单片机应用系统的应用与开发紧密结合,以针对性的课程内容和元素植入学习任务的方式,厚植爱国主义情怀和大国工匠精神,形成爱国精神引领、工匠精神铸魂、厚德载物践行的课程思政育人模式。

本书由许昌职业技术学院马建民、张松枝、刘煜辉、王宏森团队共同编写,由马建民任主编,张松枝、刘煜辉、王宏森任副主编。具体编写分工如下:项目一、项目二由刘煜辉编写,项目三由马建民编写,项目四、项目五由张松枝编写,项目六和项目七由王宏森编写。全书由张松枝审阅并统稿。感谢校企合作企业许昌中意电气科技有限公司王建民工程师的技术指导。感谢本书所列参考文献的作者,他们的工作给了我们很大的帮助和启发。感谢为本书出版付出辛勤劳动的工作人员。

尽管全体参编人员竭尽全力,但限于自身水平,书中难免会有疏漏之处,恳请广大读者不吝指正,同时欢迎您对本书内容提供宝贵建议,我们将非常感谢。

编者
2023 年 12 月

Contents

目录

项目一	单片机最小系统设计	1
1	任务一 搭建单片机最小系统	1
	任务二 单片机系统开发软件的应用	19
项目二	简易信号灯设计	36
2	任务一 8 位 LED 信号灯的单向流水灯控制	36
	任务二 霓虹灯的控制	58
项目三	定时器/计数器及中断系统应用	66
3	任务一 模拟交通信号灯的定时控制	66
	任务二 紧急情况下交通信号灯的控制	76
项目四	显示器与键盘接口技术应用	84
4	任务一 数码管显示的简易秒表设计	84
	任务二 8×8 LED 点阵显示的简易秒表设计	92
	任务三 字符型 LCD1602 广告牌的设计	96
	任务四 具有控制功能的简易秒表的设计	106
项目五	串行接口技术	117
5	任务一 彩灯的远程控制	117
	任务二 双机通信系统的设计	128

项目六

6

D/A 与 A/D 转换接口技术 136

任务一 直流电动机转速控制设计 136

任务二 简易数字直流电压表设计 144

项目七

7

单片机系统扩展设计 149

任务一 单片机扩展 RAM、EPROM 存储器设计 149

任务二 单片机串行扩展 EEPROM 存储器设计 153

任务三 简单 I/O 接口扩展设计 162

参考文献 166



项目一 单片机最小系统设计

单片机技术是现代工业自动化、电子电气、通信及物联网等领域的一门主流技术。随着我们生活及生产方式的自动化、智能化程度不断提高,单片机技术开始融入我们生活的每一个角落。几乎所有的家电内部都有单片机,无数的自动化及智能公共设施也都有单片机在工作着。同时,学习单片机技术也是学习 ARM(advanced RISC machines,高级精简指令集计算机)嵌入式系统、FPGA(field programmable gate array,现场可编程门阵列)设计等更高阶技术的基础。

本项目以 LED(light emitting diode,发光二极管)信号灯的控制引入对单片机简单控制系统的认识,并对单片机应用系统的开发过程进行介绍。本项目系统地介绍了单片机的基本概念、最小系统、51 单片机(对兼容英特尔 8051 指令系统的单片机的统称)的硬件结构、存储器配置以及数制系统等知识。教学过程中“教、学、做”相融合,达到理论与实践的统一。



课程思政

STC(宏晶)单片机是高速、低功耗、超强抗干扰的新一代 8051 单片机,被广泛应用于各个领域,如工业控制、智能家居、物联网等。

STC(宏晶)1999 年在深圳成立,其拥有自己独立的知识产权,是中国第一家击败全球单片机竞争对手的半导体公司。STC 单片机引领着行业的发展方向,是 8051 单片机全球知名品牌,是目前全球规模较大的 8051 单片机设计公司。STC 单片机的成功研发打破了国外技术的封锁,为我们国家的科技发开展辟了新的道路。

STC 单片机的成功发展是我国科技发展的一个缩影,展现了我国在科技创新领域的实力。我们应该坚定信念,积极投身于祖国的科技事业。让我们从自己做起,从身边的小事做起,去为国家的科技事业贡献自己的力量。让我们用自己的青春和智慧,为祖国的繁荣富强添砖加瓦。



任务一 搭建单片机最小系统



任务要求

利用 51 系列单片机控制一位 LED 信号灯,实现 LED 信号灯点亮控制。

任务目的

- (1) 通过一位 LED 信号灯的点亮控制,初步了解单片机应用系统的基本构成。
- (2) 了解什么是单片机和单片机最小应用系统。

任务分析

发光二极管也称为 LED,它是半导体二极管的一种,可以把电能转换成光能。直插式发光二极管工作时正极和负极两端产生的压降为 1.6~2.4 V。当发光二极管的正极到负极有一定电流通过时,它就会亮起来。使用时一般串联一个电阻,起到限流的作用,从而为发光二极管提供合适的工作电流。如图 1-1 所示,+5 V 为电源 VCC,RES 为限流电阻,LED 是发光二极管,然后接 GND(电线接地端)。如果想让发光二极管亮起来(电流在 3~20 mA 都可以点亮,假设电流为 3.3 mA),那么 RES 电阻的取值就可以很简单地计算出来。当 RES 电阻值等于 1 kΩ 时,通过发光二极管的电流为 3.3 mA,此时发光二极管就会亮起。



图 1-1 发光二极管小电路

当控制单片机 P1.0 口输出一个高电平或者低电平时,就可以控制发光二极管的亮与灭。当单片机输出高电平(输出电压为 5 V)时,加在发光二极管两端上的电压都是 5 V,没有电压差,就不会产生电流,所以发光二极管不亮。当单片机输出一个低电平时(即输出电压为 0 V),相当于接地,与直接接 GND 是一个效果,发光二极管就会被点亮。

单片机控制一位 LED 信号灯仿真电路图如图 1-2 所示。

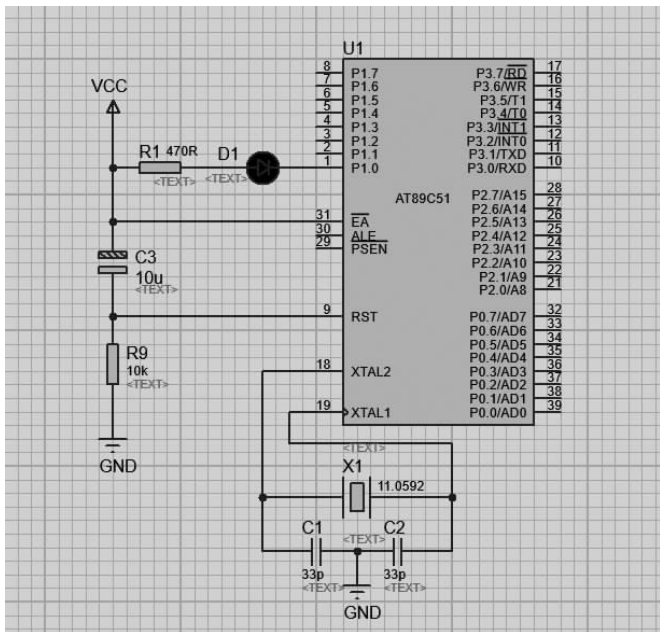


图 1-2 单片机控制一位 LED 信号灯仿真电路图

源程序设计

电路设计后,需要编写相应的控制程序。

一位 LED 信号灯点亮控制的 C 语言源程序代码如下:

```

/*****
程序名称:点亮一位 LED 信号灯
程序功能:点亮一位 LED 信号灯
*****/
#include "reg51.h"           //包含头文件 reg51.h
sbitLED=P1^0;              //位地址声明,注意:sbit 必须小写、P 大写
void main()                 //任何一个 C 语言程序都必须有且仅有一个 main 函数
{                            //{}是成对存在的,在这里表示函数的起始和结束
LED=0;                      //分号表示一条语句结束
}

```

【Proteus 设计与仿真】

在 Proteus 中运行程序,LED 将按照程序设置的方式工作。显示效果图请扫描仿真结果二维码。



单片机控制一位 LED 信号灯

相关知识

知识点 1 单片机概述

一、什么是单片机

单片微型计算机(single chip microcomputer, SCM)简称“单片机”,它不是一台机器,而是一块集成电路芯片。单片机是采用超大规模集成电路把中央处理器(central processing unit, CPU)、随机存储器(random access memory, RAM)、只读存储器(read-only memory, ROM)、中断系统、定时器/计数器、A-D 转换器、通信接口和普通 I/O 接口等集成到一块硅片上,构成的一个微型的、完整的计算机系统。单片机的 CPU 相当于个人计算机(program counter, PC)的 CPU,单片机的数据存储器相当于个人计算机的内存,单片机的程序存储器相当于个人计算机的硬盘,单片机的 I/O 接口相当于 PC 的显卡、网卡、扩展卡等的插槽……单片机可谓是“麻雀虽小,五脏俱全”。

单片机自诞生以来,以其性能稳定、低电压、低功耗、经久耐用、体积小、性价比高、控制能力强、易于扩展等优点,广泛应用于各个领域。先后出现了 4 位单片机、8 位单片机、16 位单片机、32 位单片机,在这几类单片机中最受欢迎的是 8 位单片机,它仍是目前单片机应用的主流。随着电子技术的迅速发展,单片机的功能也越来越强大。

二、单片机的发展过程

1975 年,美国德克萨斯仪器公司首次推出 4 位单片机——TMS-1000 单片机,标志着单片机的诞生。

1976年, Intel公司研制出MCS-48系列8位单片机, 使单片机的发展进入一个新阶段。MCS-48系列单片机内部集成了8位CPU、多个并行I/O接口、8位定时器/计数器、小容量的RAM和ROM等, 没有串行通信接口, 操作简单。

1980年, Intel公司在MCS-48系列单片机的基础上, 推出了MCS-51系列8位高档单片机, MCS-51系列单片机性能比MCS-48系列单片机有明显提高, 内部增加了串行通信接口, 具备多级中断处理系统, 定时器/计数器由8位扩展为16位, 扩大了RAM和ROM的容量。

1983年, 16位单片机问世, 因为性价比不理想, 并未得到普及应用, 主要应用于比较复杂的控制系统以及早期嵌入式系统。

三、单片机的应用

1. 在家用电器领域的应用

现在乃至将来, 家电将向智能化、自动化方向发展, 没有单片机智能控制的家电将被淘汰。洗衣机、电冰箱、空调机、微波炉、电饭煲、电磁炉等新型家电产品大多使用单片机进行控制。

2. 在工业自动化领域的应用

在工业控制中, 如工业过程自动控制、过程自动监测、过程数据采集、工业控制器、工业现场联网通信及机电一体化自动控制系统等都离不开单片机。在比较复杂的大型工业控制系统中, 用单片机可以实现智能控制、智能数据采集、远程自动控制、现场自动管理, 真正实现工业自动化。如工业机器人的控制系统是由中央控制器、感觉系统、行走系统、擒拿系统等节点构成的多机网络系统, 而其中每一个小系统都是由单片机进行控制的。

3. 在计算机通信领域和安全监控系统中的应用

单片机普遍具备通信接口, 可以很方便地与计算机进行数据通信, 为计算机和设备的通信提供了技术条件。如电话机及其监控设备, 楼宇自动通信呼叫系统、烟火报警系统和摄像监控系统, 无线有线对讲系统等。

4. 在医用设备领域中的应用

单片机在医用设备中的应用相当广泛, 例如医用呼吸机、分析仪、监护仪、诊断及病床呼叫系统等。

5. 在汽车电子产品中的应用

现代汽车的集中显示系统, 动力、速度、压力监测控制系统, 自动驾驶系统, 导航系统, 安全保护系统, 通信系统和运行监视器(黑匣子)等都要使用单片机。

6. 在智能仪器仪表上的应用

单片机广泛应用于仪器仪表中, 实现模拟量和数字量的转换和处理。通过传感器, 可实现诸如电压、功率、频率、温度、湿度、流量、速度、厚度、角度、长度、距离、硬度、压力、重力、音量、光照强度、波形、磁感应强度等物理量的测量。采用单片机控制, 使得仪器仪表数字化、智能化、微型化、直观化, 还能通过单片机串口通信实现远程测量和数据采集。

此外, 单片机在工商、金融、科研、教育、国防、航空航天等领域也有着广泛的应用。

单片机的广泛应用不仅让我们享受到新型电子产品和新技术带来的贴心服务, 也使我们的生活环境变得安全、舒适、便捷; 有了单片机做主控, 我们的生产生活工具变得更加先进和智能, 减轻劳动强度的同时提高了工作效率和安全系数。

知识点 2 51 系列单片机结构与引脚

想要学好单片机的应用,有必要了解并掌握单片机的组成以及单片机有哪些引脚,这些引脚都有着什么样的作用。

一、8051 单片机的基本组成

MCS-51 系列 8 位单片机因为性能可靠、简单实用、性价比高而深受欢迎,被誉为“最经典的单片机”。其中 8051 单片机是 MCS-51 系列单片机中最为典型的芯片,它只有程序存储器结构与其他型号的芯片不同,其引脚完全兼容,内部结构也完全相同。这里以 8051 单片机为例,介绍 MCS-51 系列单片机的内部组成及信号引脚。8051 单片机的内部结构如图 1-3 所示。

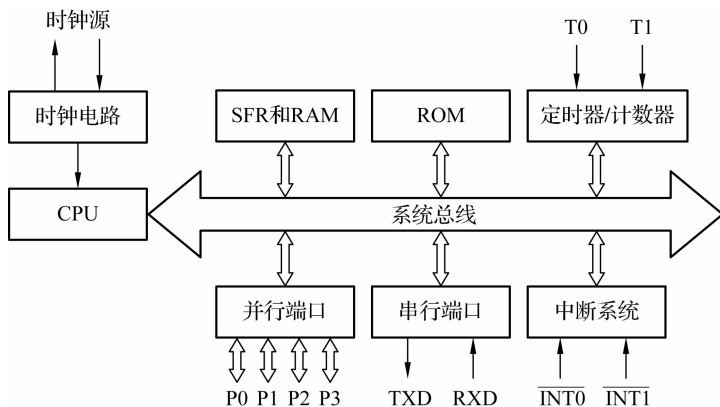


图 1-3 8051 单片机的内部结构

1. 中央处理器

中央处理器是单片机的主要核心部件,其包含了运算器、控制器以及若干寄存器组等部件。

1) 运算器

运算器是计算机的运算部件,用于实现算术运算和逻辑运算。计算机的数据运算和处理都在运算器进行。

通常运算器由算术/逻辑运算单元(arithmetic logic unit, ALU)、累加器 A、暂存寄存器、标志寄存器 F 等组成。ALU 由加法器和相应的控制逻辑电路组成。它能分别对来自两个暂存寄存器数据源的两个操作数进行加、减、与、或等运算,还能进行数据的移位。ALU 进行何种运算由控制器发出的命令确定,运算后的结果经数据总线送至累加器 A,同时影响标志寄存器 F 的状态。

(1) 累加器 A 是一个特殊的寄存器,其作用有两个:一是运算时把一个操作数经暂存寄存器送至 ALU;二是在运算后保存其运算结果。

(2) 暂存寄存器用来暂时存储数据总线或其他寄存器送来的操作数,是 ALU 的数据输入源。

(3) 标志寄存器 F 用来保存 ALU 运算结果的特征(如进位标志、溢出标志等)和处理器的状态,这些特征和状态可以作为控制程序转移的条件。

2) 控制器

计算机的控制器由指令寄存器(instruction register, IR)、指令译码器(instruction decoder, ID)、定时及控制逻辑电路和程序计数器(program counter, PC)等组成。它的控制使计算机各部分自动、协调地工作。控制器按照指定的顺序从内部程序存储器中取出指令进行译码并根据译码结果发出相应的控制信号,从而完成该指令所规定的任务。

(1)IR 用来保存当前正在执行的一条指令。要执行一条指令,首先要把它从内部程序存储器取到指令寄存器中。指令的内容包括操作码和操作数(或操作数的地址码)两部分:操作码送到 ID,经译码后确定所要执行的操作;操作数的地址码也要送到操作数地址形成电路以便形成真正的操作数地址。

(2)定时及控制逻辑电路是 CPU 的核心部件。它的任务有控制取指令、执行指令、存取操作数或运算结果等,向其他部件发出控制信号,协调各部件的工作。

(3)PC 也叫指令地址计数器。计算机的程序是有序地存储在程序存储器中的各种指令的集合。计算机运行时,按顺序取出程序存储器中的指令并逐一执行。PC 指出当前要执行的指令的地址。每当指令取出后,PC 的内容自动加 1(除转移指令外),从而指向按序排列的下一条指令的地址。若遇到转移指令(JMP)、子程序调用指令(CALL)或返回指令(RET)时,这些指令会把要执行的下一条指令的地址直接置入 PC 中,PC 的内容才会突变。PC 的位数决定了微处理器所寻址的存储器空间。

3) 寄存器组

寄存器组作为 CPU 内部的暂存单元至关重要,它是 CPU 处理数据所必需的一个存取空间,其大小直接影响着微机系统处理数据的能力和速度。

2. 内部数据存储器

8051 单片机芯片共有 256 个 RAM 单元,其中后 128 单元被专用寄存器占用,能作为寄存器供用户使用的只是前 128 单元,用于存放可读写的数。因此通常所说的内部数据存储器就是指前 128 单元,简称内部 RAM,地址范围为 00H~FFH(256 B),是一个多用多功能数据存储器,有数据存储、通用工作寄存器、堆栈、位地址等空间。

3. 内部程序存储器

8051 单片机内部有 8 KB 的 ROM,用于存放程序、原始数据或表格,因此称之为内部程序存储器,简称内部 ROM,地址范围为 0000H~FFFFH(64 KB)。

4. 定时器/计数器

8051 单片机共有两个 16 位的定时器/计数器,以实现定时或计数功能,并以其定时或计数的结果对计算机进行控制。定时时,靠内部分频时钟频率计数实现;计数时,对 P3.4(T0)或 P3.5(T1)端口的低电平脉冲计数。

5. 并行 I/O 口

8051 单片机共有 4 个 8 位的 I/O 接口(P0、P1、P2、P3)以实现数据的输入/输出。

6. 串行口

8051 单片机有一个可编程的全双工的串行口,以实现单片机与其他设备之间的串行数据传送。该串行口功能较强,既可作为全双工异步通信收发器使用,也可作为移位器使用。其 RXD(P3.0)引脚为接收端口,TXD(P3.1)引脚为发送端口。

7. 中断控制系统

8051 单片机的中断功能较强,可以满足不同控制应用的需要。8051 单片机有 5 个中断

源,即外中断 2 个、定时中断 2 个、串行中断 1 个。

8. 定时与控制部件

8051 单片机内部有一个高增益的反相放大器,其输入端为引脚 XTAL1,输出端为引脚 XTAL2。8051 单片机芯片的内部有时钟电路,但石英晶体和微调电容需外接。时钟电路为单片机产生时钟脉冲序列。

二、单片机的引脚及功能

8051 单片机是标准的 40 引脚双列直插式集成电路芯片,如图 1-4 所示。按其功能可分为电源引脚、时钟引脚、控制引脚和 I/O 接口。

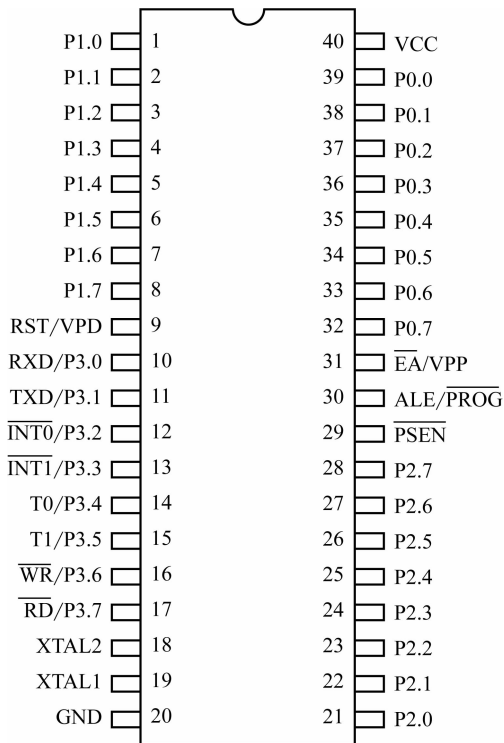


图 1-4 8051 单片机的引脚

1. 电源引脚

VCC: 芯片主电源,外接 +5 V;GND: 电源地线。

2. 时钟引脚

XTAL1 与 XTAL2 为内部振荡器的两条引出线。

3. 控制引脚

(1) ALE/ $\overline{\text{PROG}}$: 地址锁存控制信号/编程脉冲输入端。在扩展系统时,ALE 用于把 P0 口输出的低 8 位地址锁存起来,以实现低 8 位地址和数据的隔离,P0 口作为数据地址复用口线。当访问单片机外部程序或数据存储器或外接 I/O 接口时,ALE 输出脉冲的下降沿用于低 8 位地址的锁存信号;即使不访问单片机外部程序或数据存储器或外设接口,ALE 端仍以晶振频率的 1/6 输出正脉冲信号,因此可作为外部时钟或外部定时信号使用。但应注意,此时不能访问

单片机外部程序、数据存储或外设 I/O 接口。ALE 端可以驱动 8 个 TTL 负载。

(2) $\overline{\text{PSEN}}$: 片外程序存储器读选通有效信号。在 CPU 向片外程序存储器读取指令和常数时, 每个机器周期 $\overline{\text{PSEN}}$ 的两次低电平有效。但在此期间, 每当访问外部存储器或 I/O 接口时, 该 $\overline{\text{PSEN}}$ 两次低电平有效信号将不出现。 $\overline{\text{PSEN}}$ 端可以驱动 8 个 TTL 负载。

(3) $\overline{\text{EA}}/\text{VPP}$: 访问程序存储器控制信号/编程电源输入端。当该引脚 $\overline{\text{EA}}$ 信号为低电平时, 只访问片外程序存储器, 不管片内是否有程序存储器; 当该引脚为高电平时, 单片机访问片内的程序存储器。

(4) RST/VPD : 复位/掉电保护信号输入端。当振荡器运行时, 在该引脚加上一个 2 个机器周期以上的高电平信号, 就能使单片机回到初始状态, 即进行复位。掉电期间, 该引脚可接上备用电源 (VPD) 以保持内部 RAM 的数据。

4. I/O 接口

P0 口 (P0.0~P0.7): 8 位双向并行 I/O 接口。扩展片外存储器或 I/O 接口时, 作为低 8 位地址总线和 8 位数据总线的分时复用接口, 它为双向三态。

P0 口可带 8 个 TTL (transistor-transistor logic, 晶体管-晶体管逻辑电路) 负载电流。

P1 口 (P1.0~P1.7): 8 位准双向并行 I/O 接口。P1 口的每一位都可以独立设置成输入/输出位。

P1 口可以驱动 4 个 TTL 电路。

P2 口 (P2.0~P2.7): 8 位准双向并行 I/O 接口。扩展外部数据、程序存储器时, 作为高 8 位地址输出端口。P2 口可以驱动 4 个 TTL 电路。

P3 口 (P3.0~P3.7): 8 位准双向并行 I/O 接口。除了与 P1 口有一样的 I/O 功能外, 每一个引脚还兼有第二功能, 如表 1-1 所示。

表 1-1 P3 口各引脚对应的第二功能

P3.0	P3.1	P3.2	P3.3	P3.4	P3.5	P3.6	P3.7
RXD	TXD	$\overline{\text{INT0}}$	$\overline{\text{INT1}}$	T0	T1	$\overline{\text{WR}}$	$\overline{\text{RD}}$

P3 口可以驱动 4 个 TTL 电路。

P3 口的第二功能信号都是单片机的重要控制信号, 因此, 在实际使用时, 先按需要选用第二功能信号, 其余的则以第一功能的身份作为数据位的 I/O 接口使用。

P1、P2、P3 口片内均有固定的上拉电阻, 故称为准双向并行 I/O 接口; P0 口片内无固定的上拉电阻, 由两个 MOS 管串接, 既可开路输出, 又可处于高阻的“悬空”状态, 故称为双向三态并行 I/O 接口。

以上对 8051 单片机芯片上全部 40 个引脚的定义及功能做了简单的介绍。对于 MCS-51 系列各种型号的芯片, 其引脚的第一功能信号是相同的, 不同的是引脚的第二功能信号。可以对照实训电路找到相应的引脚, 在电路中查看每个引脚的连接使用。

知识点 3 单片机最小系统的组成

单片机的最小系统就是让单片机能正常工作并发挥其功能时所必需的组成部分, 也可理解为用最少的元器件组成的单片机可以工作的系统。对 MCS-51 系列单片机来说, 最小系统一般包括单片机、电源电路、时钟振荡电路、复位电路, 如图 1-5 所示。下面主要介绍后两个电路。

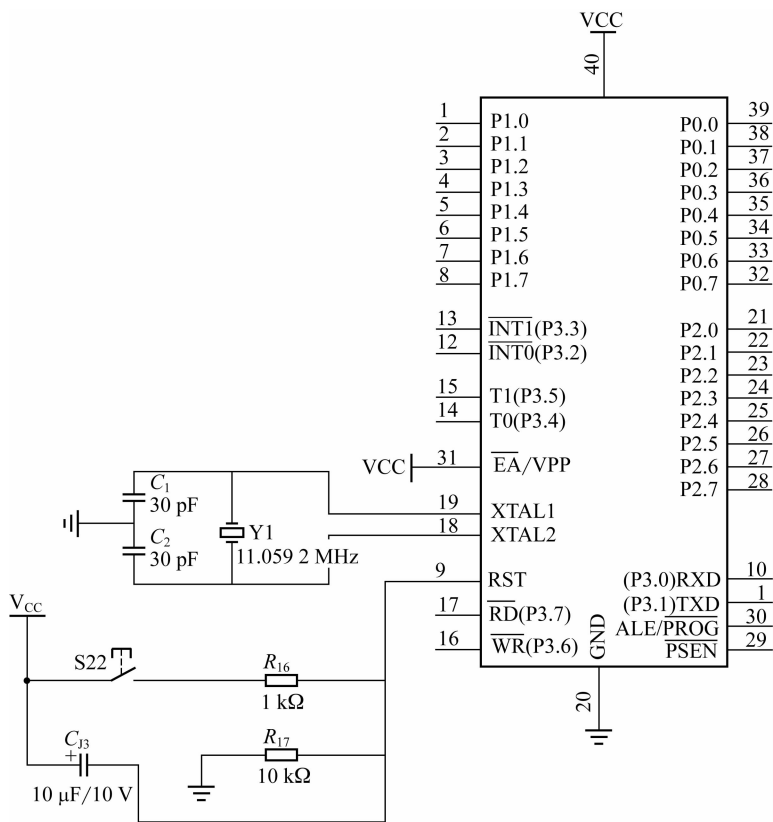


图 1-5 单片机最小系统

1. 时钟振荡电路

在 MCS-51 系列芯片内部有一个高增益反相放大器，其输入端为芯片引脚 XTAL1，输出端为引脚 XTAL2。在芯片外部，XTAL1 和 XTAL2 之间跨接晶体振荡器和微调电容，从而构成一个稳定的自激振荡器，即单片机的时钟振荡电路。电容一般在 C_1 和 C_2 取 $15 \sim 50 \text{ pF}$ ，晶体的振荡频率范围是 $1.2 \sim 12 \text{ MHz}$ 。若晶体振荡频率高，则系统的时钟频率也高，从而单片机的运行速度也就快，功耗也越大，越容易受到干扰。通常情况下，MCS-51 的应用振荡频率为 6 MHz 或 12 MHz 。

2. 复位电路

单片机复位是使 CPU 和系统中的其他功能部件都处在一个确定的初始状态，并从该状态开始工作。无论是在单片机刚开始接上电源时，还是断电后或者发生故障后都要复位。复位的作用就是让系统重新开始执行程序，也就是让程序的指针复位，回到 0000H 的位置。单片机复位操作一般有 3 种方式：上电复位、手动复位和程序自动复位。

(1) 上电复位：目的是在电路通电时马上进行复位操作，让单片机进入正常工作状态。MCS-51 系列单片机为高电平复位，通常在复位引脚 RST 上连接一个电容到 VCC，再连接一个电阻到 GND，由此形成一个 RC 充放电回路，来保证单片机在上电时 RST 脚上有持续时间超过两个时钟周期的高电平进行复位，随后回归到低电平进入正常工作状态。这个电阻和电容的典型值分别为 $10 \text{ k}\Omega$ 和 $10 \mu\text{F}$ ，如图 1-5 中的 C_{J3} 和 R_{17} 组成上电复位电路。

(2) 手动复位：目的是在必要时可以手动操作，就像计算机的重启按钮。如图 1-5 所示，

复位按键 S22 的一端接电源 VCC,另一端通过电阻 R_{16} 接到 RST 引脚,当按键按下持续时间超过两个时钟周期的高电平时进行复位,松开按键后通过 R_{17} 放电回归到低电平进入正常工作状态。

(3)程序自动复位:当单片机里的程序受到外界的干扰“跑飞”时,单片机系统通常会有一套自动复位机制,如“看门狗”电路。当程序长时间失去响应时,“看门狗”电路会自动复位单片机,让它重新工作。

知识点 4 单片机应用系统中的数制与编码

一、计数制

计数制是计数方法的统称,单片机中常用的数制有二进制、八进制、十进制和十六进制等,最常用的是十进制计数制。

1. 数的表示

$$D = a_{n-1}B^{n-1} + a_{n-2}B^{n-2} + \dots + a_1B^1 + a_0B^0$$

其中, B 为基数, B^i 为第 i 位上的位权, a_i 为第 i 位上的数码。

十进制(decimal):它有 0~9 十个不同的数码,这是构成所有十进制数的基本符号,逢十进位。例如:

$$123.45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

二进制(binary):它有 0 和 1 两个数码,基数为 2,逢二进位。例如:

$$\begin{aligned} 11010.11 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 26.75 \end{aligned}$$

十六进制(hexadecimal):它有 0~9、A~F 十六个数码,对应十进制数分别是:0~15。逢十六进位。

C 语言中十六进制数前加“0x”表示,例如:0x35。

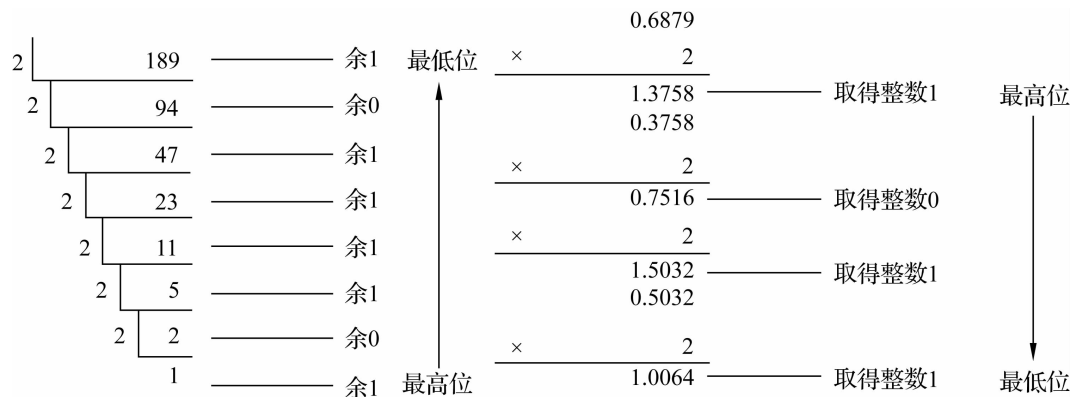
1)二进制数与十进制数间的转换

(1)二进制数转换为十进制数。例如:

$$10010.01B = 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^{-2} = 18.25$$

(2)十进制数转换为二进制数(分为十进制整数和十进制小数两种情况)。例如:

$$\begin{aligned} 1891 &= 10111101B \\ 0.6879D &\approx 0.1011B \end{aligned}$$



对于同时有整数和小数两部分的十进制数,其转换成二进制数的一种方法是把它的整数和小数部分分开转换后,再合并起来。

2) 二进制数与十六进制数间的转换

由于4位二进制数恰好有16个组合状态,即1位十六进制数与4位二进制数是一一对应的。所以,十六进制数与二进制数的转换是十分简单的。

(1) 十六进制数转换成二进制数,只要将每一位十六进制数用对应的4位二进制数替代即可——简称位分四位。例如:将 $(4AF8B)_{16}$ 转换为二进制数。

十六进制	4	A	F	8	B
二进制	0100	1010	1111	1000	1011

所以, $(4AF8B)_{16} = (1001010111110001011)_2$ 。

(2) 二进制数转换为十六进制数,从低位向高位每四位一组,依次写出每组4位二进制数所对应的十六进制数——简称四位合一位。例如,将二进制数 $(0001\ 1101\ 0110)_2$ 转换为十六进制数。

二进制	0001	1101	0110
十六进制	1	D	6

所以, $(111010110)_2 = (1D6)_{16}$

转换时注意最后一组不足4位时必须加0补齐4位。

2. 码制转换

1) 机器数与真值

机器中以编码形式表示的数称为机器数(如 $N_1 = 01011011B$ 及 $N_2 = 11011011B$)。

把原来一般书写形式表示的数称为真值($N_1 = +1011011B$ 及 $N_2 = -1011011B$)。

若一个数的所有数位均为数值位,则该数为无符号数。

若一个数的最高位为符号位而其他数位为数值位,则该数为有符号数。

由此,同一存储单元,它存放的无符号数和有符号数所能表示的数值范围是不同的。

若1个存储单元为8位,当存放无符号数时,因有效数值位为8位,故该数范围为0~255;当存放有符号数时,因有效数值位为7位,故该数范围为-127~+127。

2) 原码、反码和补码

① 原码:在符号位中用0表示正、用1表示负的二进制数,称为原码。

例如:

$$x_1 = +1110111B, [x_1]_{\text{原}} = 01110111B$$

$$x_2 = -1110111B, [x_2]_{\text{原}} = 11110111B$$

数0可是+0或-0。因此,0在原码中的形式为

$$[+0]_{\text{原}} = 00000000B, [-0]_{\text{原}} = 10000000B$$

② 反码:正数的反码=原码;负数的反码=原码的符号位不变而数值按位取反。所谓按位取反,即将各位的1变成0,0变成1。

例如:

$$x_1 = +13, [x_1]_{\text{反}} = [+13]_{\text{原}} = 00001101B$$

$$x_2 = -13, [x_2]_{\text{原}} = [-13]_{\text{原}} = 10001101\text{B}$$

$$[x_2]_{\text{反}} = [-13]_{\text{反}} = 11110010\text{B}。$$

③补码:正数的补码=原码;负数的补码=反码+1。

例如:

$$x_1 = +1101101\text{B}, [x_1]_{\text{补}} = [x_1]_{\text{原}} = 01101101\text{B}$$

$$x_2 = -1101101\text{B}, [x_2]_{\text{反}} = 10010010\text{B}$$

$$[x_2]_{\text{补}} = [x_2]_{\text{反}} + 1 = 10010011\text{B}。$$

在补码表示中,“0”是唯一的,即 $[\pm 0]_{\text{补}} = 00000000\text{B}$ 。

3)数的小数点表示方法

定点表示法:表示小数点的位置是固定不变的。分为纯整数和纯小数两类。

浮点表示法:表示小数点的位置是不固定的。任意二进制数 N 一般可表示为: $N = 2^P \times S$ 。

一个浮点数分为阶码和尾数两部分,二者各有表示正负的阶符和数符,存储格式为:
阶符 阶码 数符 数码。

4)BCD(binary coded decimal)码

BCD 码是一种用二进制编码表示十进制数的编码约定,是一种具有十进制权的二进制编码。采用标准的 8421 编码表示十进制 0~9 这十个数字。BCD 码主要用于显示、打印数据的传送等场合。

5)ASCII 码

ASCII 码是一种用八位二进制数表示字符的编码制度,是美国信息交换标准代码(American standard code for information interchange)。可以对 128 个字符进行编码(包括可显示的和不可显示的控制字符),其中最高位用于奇偶校验。扩展的 ASCII 码可以对 256 个字符进行编码。

阿拉伯数字 0~9 的 ASCII 码分别为 30H~39H。

英文大写字母 A~Z 的 ASCII 码是从 41H 开始依次往下编码。

英文小写字母 a~z 的 ASCII 码是从 61H 开始依次往下编码。

3. 位、字节、字、字长的关系

(1)位:在计算机中,一个“位”(bit,简单记为 b,也称比特)是计算机数据运输的基本单位,可以存放 0 或者 1。

(2)字节(byte):计算机中的数据大多以并行方式进行传送、存储,按 8 位一组进行,这些有序的 8 位数据就称为一个字节。有时,计算机还以双字节、四字节为单位传送数据。

(3)字(word):根据数据总线的宽度,称一次传递数据的总位数为字。如 8 位、16 位、32 位、64 位等。

在微型计算机中,通常用多少字节来表示数据传输的数量:

$$1 \text{ byte} = 8 \text{ bit}$$

$$1 \text{ KB} = 1 \times 2^{10} \text{ B} = 1024 \text{ B}$$

$$1 \text{ MB} = 1 \times 2^{10} \text{ KB} = 1024 \text{ KB}$$

$$1 \text{ GB} = 1 \times 2^{10} \text{ MB} = 1024 \text{ MB}$$

(4)字长:就是字的长度,字长被定义为 CPU 在单位时间内(同一时间)能一次处理的二进制数(字)的位数。例如一台 8 位机,它的 1 个字就等于 1 个字节,字长为 8 位。

知识点 5 单片机存储器结构

存储器的功能是存储信息(程序和数据)。存储器按其存取方式可以分成两大类,一类是随机存储器(RAM),另一类是只读存储器(ROM)。对于 RAM,CPU 在运行过程中能随时进行写入和读出,但在关闭电源时,其存储信息将丢失,所以它只能用来存放暂时性的输入/输出数据、运算的中间结果或用作堆栈。因此,RAM 常被称作数据存储器。ROM 是一种写入信息后不能改写只能读出的存储器,断电后,其信息仍保留不变。ROM 用来存放固定的程序或数据,如系统监控程序、常数表格等。所以,ROM 常被称作程序存储器。

MCS-51 单片机的芯片内部包含数据存储器和程序存储器。下面先介绍内部数据存储器。

1. 内部数据存储器的地址分配

内部数据存储器的地址分配如表 1-2 所示。

表 1-2 内部数据存储器的地址分配

字 节	地 址 分 配
F8H~FFH ⋮ 80H~87H	SFR 区
30H~7FH	用户 RAM 区(数据缓冲区)
20H~2FH	位寻址区(00H~7FH)
18H~1FH	工作寄存器区 3 区(R7~R0)
10H~17H	工作寄存器区 2 区(R7~R0)
08H~0FH	工作寄存器区 1 区(R7~R0)
00H~07H	工作寄存器区 0 区(R7~R0)

内部数据存储器共有 256 个单元,通常把这 256 个单元按其功能划分为两部分:低 128 字节(00H~7FH) 数据存储器和高 128 字节(80H~FFH) 数据存储器。

2. 内部数据存储器低 128 字节

内部数据存储器的低 128 字节(00H~7FH)是真正的内部数据存储器,按其用途划分为工作寄存器区、位寻址区和用户 RAM 区 3 个区域,如表 1-3 所示。

表 1-3 内部数据存储器低 128 字节的配置

字 节	地 址 分 配
30H~7FH	用户 RAM 区(数据缓冲区)
20H~2FH	位寻址区(00H~7FH)
18H~1FH	工作寄存器区 3 区(R7~R0)
10H~17H	工作寄存器区 2 区(R7~R0)

(续表)

字 节	地 址 分 配
08H~0FH	工作寄存器区 1 区(R7~R0)
00H~07H	工作寄存器区 0 区(R7~R0)

1) 工作寄存器区

工作寄存器区共有 4 组寄存器,每组 8 个寄存单元,各单元 8 位,每组的 8 个寄存单元都以 R7~R0 作为寄存单元的编号。寄存器常用于存放操作数及中间结果,由于它们的功能及使用不做预先规定,因此被称为通用寄存器。4 组通用寄存器占据内部数据存储器的 00H~1FH 单元地址。

在任一时刻,CPU 只能使用 4 组寄存器中的一组寄存器,把正在使用的那组寄存器称为当前寄存器组。到底是哪一组由程序状态字寄存器(PSW)中的 RS1、RS0 的状态组合来决定。

通用寄存器为 CPU 提供了就近存储数据的功能,有利于提高单片机的运算速度。此外,使用通用寄存器还能提高程序编制的灵活性,因此在单片机的应用编程中应充分利用这些寄存器,以简化程序设计,提高程序运行速度。

2) 位寻址区

内部数据存储器的 20H~2FH 单元,既可以作为一般的数据存储器单元,进行字节操作,也可以对单元中每一位进行位操作,因此把该区称为位寻址区。位寻址区共有 16 个 RAM 单元字节,共 128 位,各位地址为 00H~7FH。MCS-51 单片机具有布尔处理机的功能,位寻址区可以构成布尔处理机的存储空间。这种位寻址区能力是 MCS-51 单片机的一个重要特点,内部数据存储器位寻址区的位地址如表 1-4 所示。

表 1-4 内部数据存储器位寻址区的位地址

单 元 地 址	MSB								LSB
	位地址(十六进制)								
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	

3) 用户 RAM 区

在内部数据存储器的 128 字节中,通用寄存器占了 32 字节,位寻址区占了 16 字节,剩下 80 字节就是供用户使用的一般 RAM 区,其地址为 30H~7FH。

对用户 RAM 区的使用没有任何规定和限制,但在实际使用中,常需在用户 RAM 区设置堆栈。这在编程中使用 RAM 单元时应特别注意,不要与栈区单元混淆。

3. 内部数据存储器高 128 字节

内部数据存储器高 128 字节是供给专用寄存器使用的,其地址为 80H~FFH。但这 21 个专用寄存器的地址分散地分布在 80H~FFH 的地址空间中,只占用了高 128 字节中的 21 字节。因这些寄存器的功能已做专门规定,故称之为专用寄存器(special function register, SFR),也可称之为特殊功能寄存器。

1) 专用寄存器的类型

8051/89C51 共有 21 个专用寄存器,现对其中部分寄存器做如下介绍。

(1) 程序计数器(program counter, PC)。PC 是一个 16 位的计数器,它的作用是控制程序的执行顺序,存放下一条要执行的指令的地址,寻址范围达 64 KB。PC 有自动加 1 的功能,从而实现程序的顺序执行。PC 没有地址,是不可寻址的,因此用户无法对它进行读/写操作,但可以通过转移、调用、返回等指令改变其内容,以实现程序的转移。因地址不在 SFR 内,一般不计作专用寄存器。

(2) 累加器(accumulator, ACC)。累加器为 8 位寄存器,是最常用的专用寄存器,功能较多,地位重要。它既可用于存放操作数,也可用来存放运算的中间结果。MCS-51 单片机中大部分单操作数指令的操作数就取自累加器,许多双操作数指令中的一个操作数也取自累加器。

(3) B 寄存器。B 寄存器也是一个 8 位寄存器,主要用于乘除运算。乘法运算时, B 寄存器存乘数,乘法操作后,乘积的高 8 位存于 B 寄存器中;除法运算时, B 寄存器存除数,除法操作后,余数存于 B 寄存器中。此外, B 寄存器也可作为一般寄存器使用。

(4) 程序状态字(program status word, PSW)。程序状态字是一个 8 位寄存器,用于存放程序运行中的各种状态信息。其中有些位的状态是根据程序执行结果,由硬件自动设置的,而有些位的状态则使用软件方法设定。PSW 的位状态可以用专门指令进行测试,也可以用指令读出。一些条件转移指令根据 PSW 某些位的状态进行程序转移。PSW 的各位定义如表 1-5 所示。

表 1-5 PSW 的各位定义

D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
CY	AC	F0	RS1	RS0	OV	—	P
PSW. 7	PSW. 6	PSW. 5	PSW. 4	PSW. 3	PSW. 2	PSW. 1	PSW. 0

PSW 的字节地址为 D0H~D7H。

除 PSW. 1 位保留未用外,其余各位的定义及使用如下。

①CY(PSW. 7):进位标志位。CY 是 PSW 中最常用的标志位。其功能有:一是存放算

术运算的进位标志,在进行加减运算时,如果操作结果的最高位有进位或借位时,CY由硬件置“1”,否则清“0”;二是在位操作中作累加位使用。位传送、位与位等位操作,进位标志位是固定的操作位之一。

②AC(PSW.6):辅助进位标志位。在进行加减运算中,当低4位向高4位进位或借位时,AC由硬件置“1”,否则清“0”。在BCD码调整中也要用到AC位状态。

③F0(PSW.5):用户标志位。这是一个供用户定义的标志位,需要利用软件方法置位或复位,用于控制程序的转向。

④RS1和RS0(PSW.4和PSW.3):工作寄存器组选择位。它们被用于选择CPU当前使用的通用寄存器组。通用寄存器共有4组,其对应关系如表1-6所示。

表 1-6 通用寄存器组的选择

RS1	RS0	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

这两个选择位的状态是由软件设置的,被选中的寄存器组即为当前通用寄存器组。当单片机上电或复位后,RS1、RS0=00。

⑤OV(PSW.2):溢出(overflow)标志位。在带符号数的加减运算中,OV=1表示加减运算超出了累加器所能表示的符号数有效范围(-128~+127),即产生了溢出,因此运算结果是错误的;OV=0表示运算结果正确,即无溢出产生。

在乘法运算中,OV=1表示乘积超过255,即乘积分别在B寄存器与累加器中;OV=0表示乘积只在累加器中。

在除法运算中,OV=1表示除数为0,除法不能进行;OV=0表示除法可正常进行。

⑥P(PSW.0):奇偶标志位,表明累加器中内容的奇偶性。如果累加器中有奇数个“1”,则P置“1”,否则置“0”。凡是改变累加器中内容的指令均会影响P标志位。

该标志位对串行通信中的数据传输有重要的意义,在串行通信中常采用奇偶校验的办法来校验数据传输的可靠性。

(5)数据指针(data point register, DPTR)。数据指针为16位寄存器。编程时,DPTR既可以按16位寄存器使用,也可以按两个8位寄存器分开使用。

①DPH:DPTR高8位字节。

②DPL:DPTR低8位字节。

DPTR通常在访问外部数据存储器时作地址指针使用。由于外部数据存储器的寻址范围为64KB,故把DPTR设计为16位。

(6)堆栈指针(stack pointer, SP)。堆栈是一个特殊的存储区,用来暂存数据和地址,它是按“先进后出”的原则存取数据的。堆栈共有两种操作:进栈和出栈。

由于MCS-51单片机的堆栈设在内部RAM中,因此SP是一个8位寄存器。系统复位

后,SP 的内容为 07H,从而复位后堆栈实际上是从 08H 单元开始的,但 08H~1FH 单元分别属于工作寄存器 1~3 区,如果程序要用到这些区,最好把 SP 值改为更大的值。一般在内部 RAM 的 30H~7FH 单元中开辟堆栈。SP 的内容一经确定,堆栈的位置也就随着确定下来,由于 SP 可初始化为不同的值,因此,堆栈位置是浮动的。

2) 专用寄存器中字节寻址和位寻址

MCS-51 单片机有 21 个可寻址的专用寄存器,其中有 11 个专用寄存器是可以进行位寻址的。下面把各寄存器的字节地址及位地址列于表 1-7 和表 1-8 中。

表 1-7 专用寄存器字节地址表

符 号	名 称	地 址
ACC	累加器	E0H
B	B 寄存器	F0H
PSW	程序状态字	D0H
SP	堆栈指针	81H
DPTR	数据指针(包括 DPH 和 DPL)	82H 83H
P0	P0 口锁存寄存器	80H
P1	P1 口锁存寄存器	90H
P2	P2 口锁存寄存器	A0H
P3	P3 口锁存寄存器	B0H
IP	中断优先级控制寄存器	B8H
IE	中断允许控制寄存器	A8H
TMOD	定时器/计数器工作方式状态寄存器	89H
TCON	定时器/计数器控制寄存器	88H
TH0	定时器/计数器 0(高字节)	8CH
TL0	定时器/计数器 0(低字节)	8AH
TH1	定时器/计数器 1(高字节)	8DH
TL1	定时器/计数器 1(低字节)	8BH
SCON	串行口控制寄存器	98H
SBUF	串行口数据缓冲器	99H
PCON	电源控制寄存器	87H

表 1-8 专用寄存器位地址表

SFR	MSB								LSB	字节地址
	位地址/位定义									
B0	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	F0H	
ACC	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	E0H	
PSW	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	D0H	
	CY	AC	F0	RS1	RS0	OV	—	P		

(续表)

SFR	MSB 位地址/位定义 LSB								字节地址
IP	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	B8H
				PS	PT1	PX1	PT0	PX0	
P3	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	B0H
	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
IE	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	A8H
	EA			ES	ET1	EX1	ET0	EX0	
P2	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	A0H
	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
SCON	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	98H
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
P1	97H	96H	95H	94H	93H	92H	91H	90H	90H
	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
P0	87H	86H	85H	84H	83H	82H	81H	80H	80H
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	

对专用寄存器的字节寻址问题做如下几点说明。

(1) 21 个可字节寻址的专用寄存器不连续地分散在内部 RAM 高 128 字节之中, 尽管还有许多空闲地址, 但用户并不能使用。

(2) 程序计数器不占据 RAM 单元, 它在物理上是独立的, 因此是不可寻址的寄存器。

(3) 对专用寄存器只能使用直接的寻址方式, 书写时既可使用寄存器符号, 也可使用寄存器单元地址。

全部专用寄存器可位寻址的位共 83 位, 这些位都具有专门的定义和用途。这样, 加上位寻址的 128 位, 在 MCS-51 单片机的内部 RAM 中共有 $128+83=211$ 个可寻址位。

4. 内部程序存储器

MCS-51 单片机的程序存储器用于存放编好的程序和表格常数。8051 片内有 4 KB 的 ROM, 8751 片内有 4 KB 的可擦可编程只读存储器(erasable programmable read-only memory, EPROM), 8031 片内无 ROM。MCS-51 单片机的片外最多能扩展 64 KB ROM, 片内外的 ROM 是统一编址的。如果 EA 端保持高电平, 8051 的程序计数器在 0000H~0FFFH 地址范围内(即前 4 KB 地址)执行片内 ROM 中的程序, 当 PC 在 1000H~FFFFH 地址范围时, 自动执行片外程序存储器中的程序; 当 EA 端保持低电平时, 只能寻址外部程序存储器, 片外存储器可以从 0000H 开始编址。

MCS-51 单片机的程序存储器中有些单元具有特殊功能, 使用时应注意。

其中一组特殊单元是 0000H~0002H。系统复位后, (PC)=0000H, 单片机从 0000H 单元开始取指令执行程序。如果程序不从 0000H 单元开始, 应在这三个单元中存放一条无

条件转移指令,以便直接转去执行指定的程序。

还有一组特殊单元是 0003H~002AH,共 40 个单元。这 40 个单元被均匀地分为 5 段,作为 5 个中断源的中断地址区。

- (1)0003H~000AH 为外部中断 0 中断地址区。
- (2)000BH~0012H 为定时器/计数器 0 中断地址区。
- (3)0013H~001AH 为外部中断 1 中断地址区。
- (4)001BH~0022H 为定时器/计数器 1 中断地址区。
- (5)0023H~002AH 为串行中断地址区。

中断响应后,按中断种类,自动转到各中断区的首地址去执行程序,因此在中断地址区中理应存放中断服务程序。但通常情况下,8 个单元难以存下一个完整的中断服务程序,因此通常也是从中断地址区首地址开始存放一条无条件转移指令,以便中断响应后,通过中断地址区,再转到中断服务程序的实际入口地址。

任务二 单片机系统开发软件的应用

任务要求

实现一位 LED 信号灯的闪烁控制。

任务目的

- (1)初步了解单片机应用系统开发的过程。
- (2)掌握在 Keil μ Vision 环境中调试程序的基本方法。
- (3)了解在 Proteus 仿真软件环境中实现电路仿真应用的操作方法。

任务分析

本任务主要以单片机控制单灯闪烁为例来介绍 Keil C51 编程软件和 Proteus 仿真软件的使用方法。

信号灯应用于很多电子产品中,如电源指示灯、交通路口信号灯等。本任务中 LED 信号灯的阴极与单片机引脚 P1.0 连接,阳极通过限流电阻接高电平。当引脚 P1.0 为低电平时,LED 信号灯发光;当 P1.0 引脚为高电平时,LED 信号灯熄灭。因此,让 P1.0 每隔一定时间依次输出高、低电平信号,实现 LED 信号灯亮、灭控制,从而达到闪烁效果。

一位 LED 信号灯的闪烁控制原理图如图 1-6 所示。

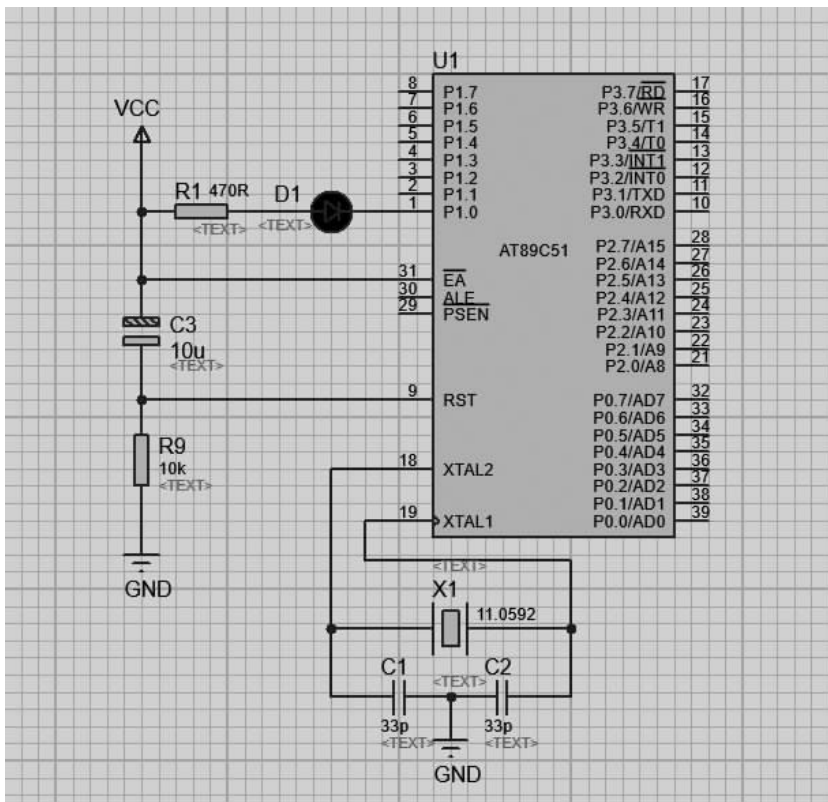


图 1-6 一位 LED 信号灯的闪烁控制原理图

源程序设计

设计电路后,需要编写相应的控制程序。

一位 LED 信号灯闪烁控制的 C 语言源程序代码如下:

```

/*****
程序名称:一位 LED 信号灯闪烁控制
程序功能:一位 LED 信号灯闪烁控制
*****/
#include "reg51.h"          //包含头文件 reg51.h
sbit LED=P1^0;             //位地址声明。注意:sbit 必须小写,P 大写
void DelayMS(uintx)       //延时
{
    uchar i;
    while(x--)
    {
        for(i=0;i<120;i++);
    }
}
    
```