



程序员书库

刘文涛 编著

01010111010

单片机应用开发实例



清华大学出版社

程序员书库

单片机应用开发实例

刘文涛 编著

清华大学出版社

北 京

内 容 简 介

本书以消防火灾自动报警器项目设计为主,从实际应用的角度介绍了单片机应用系统的开发方法、经验和技巧。主要内容包括开发前元器件的准备; Protel 在开发过程中的重要性和实际经验; 单片机开发的编程语言; 火灾自动报警器项目的设计,而这个项目设计又包括电源设计、I²C 数据巡检设计、LCD 设计和通信设计。最后还介绍了解决抗干扰问题的办法。

本书针对没有实际开发单片机经验的学生,以及刚刚走上工作岗位的技术人员。在实际开发中可能面对的问题,以实例的方式进行了阐述。

本书条理清晰、叙述简洁,适合作为高等院校学生课程设计、毕业设计及电子设计竞赛教学辅导用书,也可作为从事单片机开发的工程技术人员进行项目开发的参考书或自学用书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

单片机应用开发实例/刘文涛 编著. —北京:清华大学出版社, 2005.9

(程序员书库)

ISBN 7-302-11371-8

I.单… II.刘… III.单片微型计算机 IV.TP368.1

中国版本图书馆 CIP 数据核字(2005)第 077527 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客 户 服 务: 010-62776969

组稿编辑: 胡伟卷

文稿编辑: 刘金喜

封面设计: 王 永

版式设计: 康 博

印 装 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 22.5 字数: 519 千字

版 次: 2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷

书 号: ISBN 7-302-11371-8/TP·7485

印 数: 1~5000

定 价: 32.00 元

前 言

我国进行单片机项目开发已经有二十多年的历史，在此期间所进行的单片机开发项目已经由简单到复杂，由小型系统到大型综合系统。单片机应用也越来越广泛，从开始的工业控制，到现在的航空航天、消防安全、工业数据采集、石油地质勘探、铁路交通运输以及楼宇自动化等，甚至目前的许多家电中都有单片机的应用。随着计算机技术的飞速发展，单片机已逐渐发展成为一门关键的技术学科。

每年我国有大量的单片机相关专业的学生毕业，社会上也有相当部分的人员自学单片机知识。但是，单片机是一门非常注重实践的学科，从学习单片机理论知识，到在实际中运用单片机知识进行项目开发还有很长一段路要走，这段路就是经验的积累。而积累这些经验要付出一些代价，包括人力、材料费用以及开发周期等。所以，能了解并掌握单片机应用系统开发过程中所遇到的问题和解决办法，是在单片机开发过程中少走弯路的最好的办法，这也是编写本书的宗旨。

本书通过一个实际的单片机项目开发，来引导读者了解和掌握在实际开发过程中要运用的知识与经验。书中的各个章节安排如下：第1章介绍单片机开发前元器件的准备；第2章介绍单片机开发前软件的准备；第3章介绍 Protel；第4章介绍 C51 的编译和调试；第5章介绍楼层显示器设计；第6章介绍消防火灾自动报警器设计；第7章介绍通信设计；第8章介绍液晶显示器(LCD)设计；第9章介绍电源与抗干扰设计。

本书在编写过程中力求系统、准确、易懂，希望读者能从中受益。本书适合作为高等院校学生课程设计、毕业设计及电子设计竞赛教学辅导用书，也可作为从事单片机工作的工程技术人员进行项目开发的参考书。

本书中程序代码前标有  图标的，表示该代码可从 <http://www.tupwk.com.cn/downpage/> 上下载。

本书由刘文涛编写，其他参与创作和资料收集的人员还有：兰吉昌、王波波、赵艳、李小丽、顾正大、艾丽香、姜艳波、李长林、张玉平、刘英、聂家财、兰婵丽等。

限于作者水平，书中难免有不少缺点和错误，恳请读者批评指正。

作 者

目 录

第 1 章 单片机开发前的准备	1
1.1 单片机的任务	1
1.2 电子元件准备	3
1.2.1 电阻	3
1.2.2 电容	4
1.2.3 三极管	5
1.2.4 光耦	5
1.2.5 集成电路	6
1.2.6 继电器	6
1.2.7 电源	6
1.3 开发工具设备准备	7
1.3.1 正确使用电烙铁	7
1.3.2 仿真与仿真机	7
1.4 单片机的几个基本概念	8
1.5 开发单片机要注意的问题	10
1.5.1 提高 C 语言编程代码的效率	10
1.5.2 要注意的参数	10
1.5.3 硬件设计要注意的问题	11
1.6 单片机的抗干扰性问题	11
1.6.1 干扰的分类	12
1.6.2 抗干扰的主要方法	12
1.7 测试单片机系统的可靠性	13
1.8 选择单片机	14
1.8.1 单片机的特性	14
1.8.2 单片机的选择	15
第 2 章 软件准备	19
2.1 Keil C51	19
2.1.1 安装 Keil C51 软件	19
2.1.2 使用 Keil C51 软件	23
2.1.3 Keil C51 使用实例	26
2.2 PL/M 语言	36

2.2.1	PL/M96 语言的基本符号	38
2.2.2	标识符和保留字	38
2.2.3	高级说明语句	40
第 3 章	Protel 设计电路	42
3.1	几种常见的 EDA 介绍	42
3.1.1	Protel 软件	42
3.1.2	PADS-PowerPCB 软件	42
3.1.3	ORCAD 软件	43
3.1.4	PSPICE 软件	43
3.1.5	EWB 软件	44
3.1.6	VISIO 软件	44
3.2	Protel DXP 2004	44
3.2.1	Protel DXP 2004 的组成	44
3.2.2	Protel DXP 2004 电路板设计步骤	46
3.2.3	设计原理图	48
3.2.4	设计印制板图	64
3.2.5	Protel DXP 2004 快捷键	83
3.2.6	Protel DXP 2004 元件库封装	85
3.3	Protel 使用经验和技巧	86
3.3.1	Protel 中的一些基本概念	86
3.3.2	Protel 绘制原理图的技巧	88
3.3.3	制作印制板的技巧和要注意的问题	89
第 4 章	C51 的编译和调试	96
4.1	Keil 的编译环境 μ Vision2	96
4.1.1	μ Vision2 的窗口与菜单	98
4.1.2	编译检查工具 PC-Lint	106
4.1.3	开发工具选项	107
4.1.4	调试工具选项	113
4.2	Keil C51 编译器的控制指令	114
4.2.1	源文件控制类	114
4.2.2	目标文件(Object)控制类	114
4.2.3	列表文件(Listing)控制类	114
4.3	编译过程	115
4.3.1	A51 宏汇编器	116
4.3.2	映像文件	118
4.3.3	LIB51 库管理器	118
4.3.4	OC51 分段目标文件转换器	118

4.3.5	OH51 目标代码到 HEX 文件的转换器	119
4.4	代码优化	119
4.4.1	优化级别	119
4.4.2	对 8051 的特殊优化	119
4.5	混合编译	120
4.5.1	项目目标和文件组	120
4.5.2	浏览项目窗口中的文件和文件组的属性	120
4.5.3	用户上电初始化程序 STARTUP.A51 简介	121
4.5.4	与汇编语言的接口	124
4.5.5	与 PL/M51 的接口	130
4.6	μ Vision2 的其他功能	130
4.6.1	多个文件中查找	130
4.6.2	资源浏览器	131
4.7	出错信息	131
4.8	Keil C51 的调试与仿真	132
4.8.1	μ Vision2 调试器的几项功能	133
4.8.2	断点	134
4.8.3	仿真	136
4.8.4	μ Vision2 的调试命令	147
4.9	RTX-51 实时操作系统	150
4.9.1	概述	150
4.9.2	RTX-51 Tiny 的例程 TRAFFIC.C	156
第 5 章	楼层显示器设计	166
5.1	总体设计	166
5.2	硬件设计	167
5.2.1	选择单片机	167
5.2.2	显示系统设计	172
5.2.3	键盘系统设计	174
5.2.4	电源系统设计	175
5.2.5	复位电路设计	176
5.3	软件设计	180
5.3.1	A/D 转换部分程序	182
5.3.2	时钟日历芯片部分程序	183
5.3.3	字符液晶显示程序	192
第 6 章	消防火灾自动报警器设计	196
6.1	硬件设计	196
6.1.1	选择单片机	197

6.1.2	8279 键盘显示管理芯片	197
6.1.3	键盘显示系统设计	203
6.1.4	火警巡检系统设计	207
6.1.5	总体设计电路	209
6.2	软件设计	209
6.2.1	声明变量	210
6.2.2	巡检程序	214
6.2.3	A/D 转换子程序	219
6.2.4	键盘显示程序	223
6.2.5	高速输入中断子程序	247
第 7 章	通信设计	248
7.1	报警器与 CRT 通信设计	248
7.2	报警器与楼层显示器设计	270
7.2.1	单片机双机通信技术	270
7.2.2	单片机多机通信技术	272
第 8 章	液晶显示器(LCD)设计	276
8.1	12232 型 LCD 应用设计	276
8.1.1	12232 型 LCD 与 51 系列单片机接口	276
8.1.2	软件编程	277
8.2	12864 型 LCD 应用设计	281
8.2.1	12864 型 LCD 接口	282
8.2.2	软件设计	283
8.3	240128 型 LCD 应用设计	302
8.3.1	240128 型 LCD 接口	302
8.3.2	软件设计	303
第 9 章	电源与抗干扰设计	321
9.1	电源设计	321
9.2	抗干扰设计	326
9.2.1	硬件考虑	326
9.2.2	软件考虑	330
附录 A	Keil C51 库函数	334
附录 B	Keil C51 编译致命错误信息	338
附录 C	Keil C51 编译语法错误信息	341

第 1 章 单片机开发前的准备

单片机的全称为单片微型计算机(Single Chip Microcomputer), 它是把组成微型计算机的各功能部件, 如中央处理器(CPU)、随机存取存储器(RAM)、只读存储器(ROM)、I/O 接口电路、定时/计数器, 以及串行通信接口等部件制作在一块集成芯片中, 构成一个完整的微型计算机。而单片机的开发, 就是充分利用单片机的这些功能部件的特性实现其功能。

1.1 单片机的任务

单片机的任务是指以单片机为核心, 构建硬件部分和软件部分组成, 配以一定的外围电路和软件, 实现某几种功能, 完成相应的任务。硬件是系统的基础, 软件则是在硬件的基础上对其合理的调配和使用, 从而完成应用系统所要完成的任务。一般来讲, 所要完成的任务不同, 相应的硬件配置和软件配置也就不同。因此, 单片机的设计应包括硬件设计和软件设计两大部分。

单片机硬件设计包括两大部分内容。一是单片机系统的扩展部分设计, 它包括存储器扩展和接口扩展。存储器的扩展指 EPROM、EEPROM 和 RAM 的扩展, 接口扩展是指 8255、8155、8279 以及其他功能器件的扩展。二是各功能模块的设计, 如信号测量功能模块、信号控制功能模块、人机对话功能模块、通信功能模块等, 根据系统功能要求配置相应的 A/D、D/A、键盘、显示器、打印机等外围设备。

在进行硬件设计时, 首要问题是确定电路的总体方案, 并需进行详细的技术论证。所谓硬件电路的总体设计, 即是为实现该项目全部基本功能所需要的所有硬件的电气连线原理图。初次接触这方面工作的设计人员, 往往急于求成而在设计总体方案上不愿花更多的时间, 过于急促地开始制板和调试。这种方法不仅不妥当, 而且往往是得不偿失。因为就硬件系统来讲, 电路的各部分都是紧密相关、互相协调的, 任何一部分电路考虑得不充分, 都会给其他部分带来难以预料的影响, 轻则使系统整体结构受破坏, 重则导致硬件总体大返工, 由此造成的后果是可想而知的。所以, 我们希望设计者不要吝啬在总体方案上所花的时间。从时间上看, 硬件设计的绝大部分工作量往往在最初方案的设计阶段, 一个好的设计方案往往会有事半功倍的效果。一旦总体方案确定下来, 下一步的工作就会进行得很顺利, 即使需要作部分修改, 也只是在此基础上进行一些完善工作, 而不会造成整体返工。

在硬件设计的过程中要用到 Protel 等印制板辅助设计软件, 先用 Protel 将硬件连线原理图画好, 然后再用 Protel 将原理图转换为印制板图即 PCB 图, 检查无误后将 PCB 图交给印制板生产厂家, 印制板生产好后, 将元器件焊接到印制板上就完成了硬件系统的设计。在硬件系统设计过程中, 一定要仔细检查原理图是否正确, 生产厂家

提供的印制板是否合格。

在进行总体设计时,软件设计和硬件设计应统一考虑,结合进行。当系统电路设计定型后,软件的任务也就明确了。

系统中的应用软件是根据系统功能要求设计的。一般来讲,软件的功能可分为两大类:一类是执行软件,它能完成各种实质性的功能,如测量、计算、显示、打印、输出控制等;另一类是监控软件,它专门用来协调各执行模块和操作者的关系。

一个单片机应用系统经过总体设计、硬件设计、软件设计、制板、元器件安装后,在系统的程序存储器中放入编制好的应用程序,系统即可运行。但一次性成功几乎是不可能的,多少会出现一些硬件、软件上的错误,这就需要通过调试来发现错误并加以改正。由于单片机在执行程序时是无法用人工控制的,因此为了能调试程序,检查硬件、软件运行状态,就必须借助某种开发工具模拟用户实际的单片机,并且能随时观察运行的中间过程而不改变运行中的数据性能和结果,从而进行模仿现场的真实调试。完成这一仿真工作的开发工具就是单片机仿真器。

单片机仿真器必须具有以下基本功能。

- 能输入和修改用户的应用程序;
- 能对用户系统硬件电路进行检查和诊断;
- 能将用户源程序编译成目标码并固化到 EPROM 中;
- 能以单步、断点、连续方式运行用户程序,正确反映用户程序执行的中间结果。

对于一个完善的仿真系统,为了方便用户调试,提高产品的开发效率,还应具备以下特点:

(1) 不占用用户单片机的任何资源,包括 8031 内部 RAM、特殊功能寄存器、I/O 口、串行口及中断源等;

(2) 能提供给用户足够的仿真 RAM 空间作为用户的程序存储器(最好是从零地址开始),并提供给用户足够的 RAM 空间作为用户的数据存储器;

(3) 可以以单步、断点、全速断点、连续方式运行仿真 RAM 或样机 EPROM 内的用户程序;

(4) 有较齐全的软件开发工具,如配备有交叉汇编软件,将用户用汇编语言编制的应用程序生成可执行的目标文件;具有丰富的子程序库,汇编时连同用户设计的程序一起编译成目标程序,装入仿真 RAM 供调试和固化;具有高级语言编译系统,用户可用 BASIC 语言或 C51 语言进行编程;具有反汇编功能,对目标程序反汇编的结果可以打印或写入磁盘等。

单片机应用系统的设计包括下述几个步骤:

- (1) 总体设计。
- (2) 系统硬件设计(用 Protel)。
- (3) 系统软件设计(用仿真机软件)。
- (4) 仿真调试硬件和软件(用仿真机)。
- (5) 固化应用程序(用仿真机)。
- (6) 脱机运行(用户系统)。

1.2 电子元件准备

一个完整的单片机系统，除了单片机作为控制器的核心外，有时候还由很多元器件构成。常见的元器件包括电阻、电容、三极管、继电器、光电耦合器、集成电路芯片以及必须的电源设备等。

1.2.1 电阻

1. 电阻的种类

电阻种类按照工艺可以分为碳膜电阻和金属膜电阻，按照功率可以分为小功率电阻和大功率电阻。大功率电阻通常是金属电阻，功率一般在 10W 以上。

电阻在电路中起到限流、分压等作用。常用的是 1/8W 电阻。但电流大的地方，要考虑加大电阻的功率，可用 1/4W、1/2W 电阻等。

2. 电位器

电位器就是可调电阻。电位器分单圈和多圈电位器。单圈电位器通常为灰白色，面上有一个“十”字可调的旋纽，出厂前放在一个固定的位置上。多圈电位器通常为蓝色，调节的旋纽为“一”字。通常模拟电路中会有一些不确定的因素，需要调节才能达到理想的效果。

3. 排电阻

排电阻通常是一排 n 个电阻的封装，比较常用的就是阻值 502 和 103 的 9 脚的电阻排，像 sip9 就是 8 个电阻封装在一起，8 个电阻有一端连在一起，为公共端，在排电阻上用一个小点表示。排电阻通常为黑色或黄色。

51 单片机系统的 P0 口需要一个排电阻上拉，否则，作为输入的时候，不能正常读入数据。

4. 光敏电阻

光敏电阻是当照在电阻上的光强变化时，电阻值也发生变化。使用光敏电阻可以检测光强的变化。

1.2.2 电容

1. 电容的主要种类

电容有以下几大类：

- 电解电容；
- 独石电容；
- 磁片电容；
- 钽电解电容；
- 涤纶电容等。

电容的指标是耐压值和电容容量。例如：220 μ F/50V，是指这个电解电容耐压值为 50V，容量为 220 μ F。电容的容量跟电容的介质有关。

电解电容使用电解质作为介质，铝作为电极；独石电容使用独石作为介质；磁片电容使用磁片作为介质；钽电解电容使用电解质作为介质，但是，电极采用钽金属；涤纶电容采用涤纶作为介质。

2. 电容的使用场合

(1) 电源稳压和滤波

电解电容主要用来稳压和低频交流滤波，高频滤波使用磁片电容和独石电容。

当电解电容作为稳压时，接在整流桥和三端稳压器的输出端，起到稳定电压的作用。这时候的电解电容通常采用的容量很大，在设计时要留有余量，保证系统正常工作到它的寿命。

有些远端供电的直流电源，接到电路板的输入端时，也需要在电路板的电源输入端加一个大的电解电容，通常可以是 220 μ F/25V。这样，这块电路板需要供电时，不是直接从电源处取，而是从电容中取电，这样可以得到稳定的电流供给。

电解电容只能滤除低频的波动，对于直流电源中的高频波动，可以加一个 0.1 μ F 或 0.01 μ F 的独石电容或者磁片电容。在每一个芯片的电源和地两端接一个 0.1 μ F 或 0.01 μ F 的独石电容或者瓷片电容，可以解决芯片的供电过程中，由于电路板的走线电感产生的电源开关噪声尖峰。这种作用下的电容叫去耦电容。这是电路板的常规设计。

(2) 定时参数

对于像 555 这样需要外接电容产生稳定脉冲的器件，涤纶电容是首选。涤纶电容受到温度变化引起的变化要远远小于独石电容。

(3) 产生其他电压

有些需要从单一电压产生其他电压的芯片，如 max232，需要外接电容才能实现。

1.2.3 三极管

1. 三极管的4种工作状态

三极管有4种工作状态：

- 饱和导通状态；
- 截止状态；
- 线性放大状态；
- 非线性工作状态。

2. 三极管的具体应用

常用的PNP三极管是2N5551，电压为24V，驱动40mA的LED、蜂鸣器等。

实际上，通常采用7407、ULN2003取代三极管。但受到PCB面积的制约，以及为了降低成本，同时为布局方便，在一两个输出点时，还是可以使用三极管来做驱动的。

1.2.4 光耦

光耦是用来隔离输入/输出信号的器件。在应用中，往往有一些远距离的开关量信号需要传送到控制器，如果直接将这些信号接到单片机的I/O上，有以下的问题：

- 信号不匹配，输入的信号可能是交流信号、高压信号、按键等干接点信号；
- 比较长的连接线路容易引进干扰、雷击、感应电等，必须隔离。

常见的光耦有：

- TLP521-1/ TLP521-2/ TLP521-4，分别是1个光耦、2个光耦和4个光耦，HP公司和日本的东芝公司生产。
- 4N25/4N35，Motorola公司生产。
- 6N136，HP公司生产。

光耦的隔离电压高达5 000V。选择光耦看使用场合，TLP521-1是最常用的，也便宜。要求隔离电压高的，选用4N25/4N35。要求在通信中高速传输数据的，选用6N136。

光耦常用于：

- 输入干接点隔离；
- 输入TTL电平隔离；
- 输入交流信号隔离；
- 输出RS232信号隔离；
- 输出RS422信号隔离。

1.2.5 集成电路

1. 74 系列芯片

目前的系统中,有些 74 系列芯片经常使用,例如 7404、7407、74138 以及 74373 等。每种芯片可以根据数据手册查它的真值表来确定其功能和具体使用方法。

通常用 74LS244/74LS245/74LS373/74LS573 加强驱动能力和数据缓冲能力,74LS138 用作三-八译码器。

2. CD4000 系列芯片

CD4000 系列的芯片,与 74 系列的电气特性区别为:CD4000 电压范围宽,可以在 3V~15V 工作,输入阻抗高,驱动能力差。其主要特性为:

- (1) 输入时,1/2 工作电压以下为 0,1/2 工作电压以上为 1;输出时,1=工作电压;0=0V;
- (2) 驱动能力差,在设计时最多只能带 1 个 TTL 负载;
- (3) 上拉电阻至少要 100k Ω 。

1.2.6 继电器

继电器有公共端、常开端、常闭端。继电器的主要指标有工作电压、开关次数和功率。

(1) 工作电压

工作电压有直流的和交流的两种。通常单片机驱动直流继电器。交流的继电器通常为 AC24V 和 AC220V。直流的继电器通常为 DC5V/DC9V/DC12V/DC24V 等。

(2) 种类

根据触点数量,继电器可以分为:单刀单掷/SPST、单刀双掷/SPDT、双刀双掷/DPDT 以及四刀双掷等。

1.2.7 电源

单片机通常采用开关电源,它是将 AC220V 直接整流滤波成高压,推动功率管工作在 100kHz 以上,再整流滤波成低压。

也有采用线性电源的,用变压器将 AC220V 变换成低压,经过全桥的整流和大电容的滤波,成为脉动的直流。再经过三端稳压器,输出直流电压。

线性电源输出端还需要大的电解电容滤波,要选用足够大的电容容量和耐压值的电解电容。电容容量不够,输出电压会有纹波,单片机系统工作会不正常。

通常单片机电路板上需要 5V 直流电压,所以电源电路板采用 7805 以及加散热片。

设计时要注意,7805 的效率为 50%,变压器的效率为 50%,也就是说,用 4W 的变压器,最后最大的输出是 2W。如果需要大功率的电源供给,可以购买成品的开关电源或者线性电源。

1.3 开发工具设备准备

1.3.1 正确使用电烙铁

焊接技术是开发单片机必须掌握的一项基本技术,需要多多练习才能熟练掌握。

(1) 选用合适的焊锡,应选用焊接电子元件用的低熔点焊锡丝。

(2) 助焊剂,用 25%的松香溶解在 75%的酒精(重量比)中作为助焊剂。

(3) 电烙铁使用前要上锡。具体方法是:将电烙铁烧热,待刚刚能熔化焊锡时,涂上助焊剂,再用焊锡均匀地涂在烙铁头上,使烙铁头均匀地“吃”上一层锡。

(4) 焊接方法:把焊盘和元件的引脚用细砂纸打磨干净,涂上助焊剂。用烙铁头沾取适量焊锡,接触焊点,待焊点上的焊锡全部熔化并浸没元件引线头后,电烙铁头沿着元器件的引脚轻轻往上一提离开焊点。

(5) 焊接时间不宜过长,否则容易烫坏元件,必要时可用镊子夹住引脚帮助散热。

(6) 焊点应呈正弦波峰形状,表面应光亮圆滑,无锡刺,锡量适中。

(7) 焊接完成后,要用酒精把线路板上残余的助焊剂清洗干净,以防炭化后的助焊剂影响电路正常工作。

(8) 集成电路应最后焊接,电烙铁要可靠接地,或断电后利用余热焊接。或者使用集成电路专用插座,焊好插座后再把集成电路插上去。

(9) 电烙铁应放在烙铁架上。

1.3.2 仿真与仿真机

仿真是单片机开发过程中非常重要的一个环节,除了一些极简单的任务,一般产品在开发过程中都要进行仿真。仿真的主要目的是进行软件调试,当然借助仿真机,也能进行一些硬件排错。

一块单片机应用电路板包括单片机部分及为达到使用目的而设计的应用电路,仿真就是利用仿真机来代替应用电路板(称目标机)的单片机部分,对应用电路部分进行测试、调试。

仿真有 CPU 仿真和 ROM 仿真两种。所谓 CPU 仿真是指用仿真机代替目标机的 CPU,由仿真机向目标机的应用电路部分提供各种信号、数据,进行调试的方法。这

种仿真可以通过单步运行、连续运行等多种方法来运行程序，并能观察到单片机内部的变化，便于改正程序中的错误。所谓 ROM 仿真，就是用仿真机代替目标机的 ROM，目标机的 CPU 工作时，从仿真机中读取程序并执行。这种仿真其实就是将仿真机当成一片 EPROM，只是省去了擦片、写片的麻烦，并没有多少调试手段可言。通常这是两种不同类型的仿真机，也就是说，一台仿真机不能既作 CPU 仿真，又作 ROM 仿真。可能的情况下，当然以 CPU 仿真为好。

1.4 单片机的几个基本概念

单片机中有一些基本的概念，许多书籍和资料并没有深入地讲解这些概念，但这些内容在开发单片机中是必须要理解的。下面介绍这些最基本的概念。

1. 总线

计算机电路是以微处理器为核心的，各器件都要与微处理器相连，各器件之间的工作必须相互协调，所以需要很多的连线。如果在微处理器和各器件间单独连线，线的数量将多得惊人，所以在微处理机中引入了总线的概念，各个器件共享用连线，所有器件的数据线全部接到公用的线上，即相当于各个器件并联起来。如果有两个器件同时送出数据，一个为 0，一个为 1，这种情况是不允许的，所以要通过控制线进行控制，使器件分时工作，任何时候只能有一个器件发送数据。器件的数据线称为数据总线，器件所有的控制线称为控制总线。在单片机内部或者外部存储器及其他器件中有存储单元，这些存储单元要被分配地址才能使用，分配地址当然也是以电信号的形式给出的，由于存储单元比较多，所以用于地址分配的线也较多，这些线称为地址总线。

2. 数据、地址、指令

之所以将这三者放在一起，是因为这三者的本质都是一样的——数字，或者说都是一串 0 和 1 组成的序列。换言之，地址、指令也都是数据。

- 指令：由单片机芯片的设计者规定的一种数字，它与我们常用的指令助记符有着严格的一一对应关系，不可以由单片机的开发者更改。
- 地址：是寻找单片机内部、外部的存储单元和输入/输出口的依据。内部单元的地址值已由芯片设计者规定好，不可更改。外部单元的地址值可以由单片机开发者自行决定，但有一些地址单元是一定要有的(详见程序的执行过程)。
- 数据：这是由微处理机处理的对象，在各种不同的应用电路中各不相同，一般而言，被处理的数据可能有以下几种情况。
 - ◆ 地址，如 `MOV DPTR, #1000H`，即地址 1000H 送入 DPTR。
 - ◆ 方式字或控制字，如 `MOV TMOD, #3`，3 即是控制字。
 - ◆ 常数，如 `MOV TH0, #10H`，10H 即定时常数。

- ◆ 实际输出值(如 P1 口接彩灯, 要灯全亮, 则执行指令: MOV P1, #0FFH, 要灯全暗, 则执行指令: MOV P1, #00H), 这里 0FFH 和 00H 都是实际输出值。又如用于 LED 的字形码, 也是实际输出的值。

理解了地址、指令的本质, 就不难理解程序运行过程中为什么会跑飞, 会把数据当成指令来执行了。

3. P0 口、P2 口和 P3 口的第二功能用法

初学时往往对 P0 口、P2 口和 P3 口的第二功能用法迷惑不解, 认为第二功能和原功能之间要有一个切换的过程, 或者说要有一条指令。事实上, 各端口的第二功能完全是自动的, 不需要用指令来转换。如 P3.6、P3.7 分别是 WR、RD 信号, 当单片机外接 RAM 或有外部 I/O 口时, 它们被用作第二功能, 不能作为通用 I/O 口使用, 只要单片机一执行到 MOV X 指令, 就会有相应的信号从 P3.6 或 P3.7 送出, 不需要事先用指令说明。事实上“不能作为通用 I/O 口使用”也并不是“不能”而是使用者“不会”将其作为通用 I/O 口使用。你完全可以在指令中安排一条 SETB P3.7 的指令, 并且当单片机执行到这条指令时, 也会使 P3.7 变为高电平, 但使用者不会这么去做, 因为通常这会导致系统的崩溃(即死机)。

4. 程序的执行过程

单片机在通电复位后 8051 内的程序计数器(PC)中的值为 0000, 所以程序总是从 0000 单元开始执行。也就是说, 在系统的 ROM 中一定要存在 0000 这个单元, 并且在 0000 单元中存放的一定是一条指令。

5. 堆栈

堆栈是一个区域, 是用来存放数据的, 这个区域本身没有任何特殊之处, 就是内部 RAM 的一部分, 特殊的是它存放和取用数据的方式, 即所谓的“先进后出, 后进先出”, 并且堆栈有特殊的数据传输指令, 即 PUSH 和 POP。有一个特殊的专为其服务的单元, 即堆栈指针 SP, 每当执行一次 PUSH 指令时, SP 就在原来值的基础上自动加 1, 每当执行一次 POP 指令, SP 就在原来值的基础上自动减 1。由于 SP 中的值可以用指令加以改变, 所以只要在程序开始阶段更改了 SP 的值, 就可以把堆栈设置在规定的内存单元中。如在程序开始时, 用一条 MOV SP, #5FH 指令; 就是把堆栈设置在从内存单元 60H 开始的单元中。一般程序的开头总有这么一条设置堆栈指针的指令, 因为开机时, SP 的初始值为 07H, 这样就使堆栈从 08H 单元开始, 而 08H 到 1FH 这个区域正是 8031 的第二、三、四工作寄存器区, 经常要被使用, 这会造成数据的混乱。不同的程序员编写程序时, 初始化堆栈指令也不完全相同, 这是习惯问题。当设置好堆栈区后, 并不意味着该区域成为一种专用内存, 它还是可以像普通内存区域一样使用, 只是一般情况下编程者不会把它当成普通内存使用了。