

面向 21 世纪高等学校规划教材

电子信息与电气学科规划教材·电子信息科学与工程类专业

单片机原理与应用

系统设计

欧伟明 何静 凌云 刘剑 等编著
肖伸平 主审



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

电子信息与电气学科规划教材·电子信息科学与工程类专业

单片机原理与应用系统设计

欧伟明 何 静 凌 云 刘 剑 等编著

肖伸平 主审

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以89S51为典型机,主要论述单片机的基本结构与工作原理,以及单片机应用系统的设计与开发方法。全书内容分为13章,涉及概述、单片机的结构和工作原理、指令系统、单片机程序设计基础、中断系统、定时器/计数器、串行口 UART、单片机的常用接口技术、串行总线技术、单片机应用系统的开发环境、基于嵌入式实时操作系统的单片机程序设计方法、基于 RTX51 的乐曲编辑器和发生器的设计、数控电流源的设计。书后附录给出了18个单片机课程设计课题,以及 AT89S51 的指令系统。本书从工程应用出发,突出了单片机应用技术的新颖性和实用性;此外,本书为任课教师免费提供电子课件。

本书可作为高等学校“单片机原理与应用”、“微机原理与接口技术”课程的教材,也可供从事单片机应用系统开发的工程技术人员参考,还可作为各类电子设计竞赛的培训教材,以及单片机课程设计的参考书和电类专业学生毕业设计的参考书。

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

单片机原理与应用系统设计/欧伟明等编著. -北京:电子工业出版社,2009.7

电子信息与电气学科规划教材·电子信息科学与工程类专业

ISBN 978-7-121-08836-0

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

中国版本图书馆 CIP 数据核字(2009)第 075051 号

责任编辑:谭海平

印 刷:北京天宇星印刷厂

装 订:三河市万和装订厂

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 印张:20.25 字数:583 千字

印 次:2009 年 7 月第 1 次印刷

印 数:5000 册 定价:35.00 元

凡所购买电子工业出版社的图书有缺损问题,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。

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。

服务热线:(010)88258888。

前 言

自 1971 年微型计算机问世以来,由于实际应用的需要,微型计算机向着两个方向发展:一个是向着高速度、大容量、高性能的高档微机方向发展;而另一个则是向着稳定可靠、体积小、功耗低、价格低廉的单片机方向发展。单片机是微型计算机的一个重要分支,它的出现是计算机技术发展史上的一个重要里程碑,它使计算机从海量存储与高速复杂数值计算进入到智能化控制领域。从此,计算机技术的两个重要领域——通用计算机领域和嵌入式计算机领域都获得了极其重大的进展。

单片机诞生于 20 世纪 70 年代,美国 Intel 公司自 1976 年宣布并于 1977 年推出 MCS-48 单片机以来,世界上的单片机技术已经走过了 30 余年的历程。我国自 20 世纪 70 年代末 80 年代初就开始进行单片机的应用与开发工作,1987 年 10 月 27 日,在上海成立中国微计算机单片机学会,中国的单片机开发与应用经历了 20 余年。几十年来,单片机不是以其位数的高低来决定单片机的先进性的,而是以如何适合千变万化的应用产品的需求、高性能价格比的配置来决定单片机的优劣的。因此,高性价比、多功能、低功耗的 8 位单片机一直是单片机的主角。

随着计算机技术、大规模集成电路技术、软件技术的飞速发展,单片机技术也在快速更新。为了将当前的单片机应用技术反映到本书中,而本书又能适合教师的教学工作,以及符合学生的学习规律,在编著本书时,我们主要考虑了以下几点。

(1) 关于单片机的选型

目前,国内外公认的单片机标准体系结构是美国 Intel 公司的 MCS-51 系列,其中的 8051 单片机由 Intel 公司以技术转卖的方式,被许多半导体生产厂家作为基核,发展了许多兼容系列,所有这些系列统称为 80C51 系列,因此,人们在设计单片机应用系统时,可以根据应用系统的要求,广泛选择最佳型号的单片机。然而,美国 Atmel 公司的单片机 AT89S51 是 80C51 系列的典型代表,所以本书以 89S51 芯片为主线,介绍单片机的原理及应用。

(2) 全书整体架构

全书分为两大部分。

第一部分,即本书的第 1~8 章,主要介绍单片机结构原理及基本应用,是继续学习单片机应用技术的基础,是“单片机原理与应用”课程的经典内容。

第二部分,即本书的第 9~13 章,主要介绍当前的单片机应用新技术,以及单片机应用系统设计开发的方法和工程设计实例。很显然,第二部分是“单片机原理与应用”课程经典内容的扩展,主要目的是想给读者提供继续学习和掌握单片机应用系统开发技术的精选内容,让读者了解当前单片机应用新技术、开发小工具、开发环境、开发过程,从而达到初步掌握单片机应用系统设计与开发技术的目的。

(3) 精心安排“经典内容”,认真撰写第 1~8 章

从工程实用的角度出发,通过精心安排本书第 1~8 章的内容,达到既能讲透单片机的结构原理,又能精简“经典内容”体系结构的目的。比如,单片机的工作方式有很多种,但从工程应用的角度来看,主要用到复位工作方式、低功耗工作方式和编程工作方式,本书在介绍复位电路的基础上,重点介绍复位工作方式、低功耗工作方式和 AT89S51 的 ISP 编程工作方式,而对其他工作方式只略为提一下。又比如,定时器/计数器有 4 种工作方式,其他教材一般是顺序介绍工作方式 0 至

工作方式3,但是在实际的工程应用中,工作方式0很少采用,而工作方式1应用最多,所以本书按照工作方式1、工作方式2、工作方式3、工作方式0的顺序进行介绍,从而可以使教材内容贴近实际需要。又比如,单片机的存储器有程序存储器和数据存储器,而实际上特殊功能寄存器也属于单片机的存储器,所以本书将这3个方面的内容放在1个小节内进行介绍,这种安排是有别于其他教材的。为了保证本书中所采用的实例程序的正确性,我们做到了所选用的实例程序都是通过了实际验证的正确程序。

(4) 详细介绍串行总线技术

随着计算机技术和半导体技术的发展,MCU芯片内部资源越来越丰富,总线型单片机的非总线应用模式使用得越来越广泛,在MCU的外部,很少采用并行三总线(数据总线、地址总线、控制总线)的结构,人们常常采用具有串行总线接口的外围芯片,因此,本书用一章的篇幅来详细介绍串行总线技术,包括RS-232C、RS-485、SPI、I²C、CAN、USB、1-Wire单总线等串行总线技术。

(5) 介绍C51高级语言程序设计方法

目前,在单片机程序设计部分,讲授内容大多限于“汇编语言”,而在实际应用中,单片机程序在多年前就进入了“高级语言”的阶段,各种单片机高级语言开发工具的相继出现,使得高级语言程序设计无论在可读性、可靠性和编程效率上都远远地超过了汇编语言,德国Keil Software公司的Keil C51编译器就是典型代表。Keil C51编译器是一种专为MCS-51系列单片机应用开发而设计的高效率C语言编译器,该编译器包括C51交叉编译器、A51宏汇编器、BL51连接定位器和基于Windows的集成化文件管理编译环境、多视窗软件仿真调试器等一系列开发工具,具有高效、可靠、使用方便等许多优点,其应用如今已十分普及。

尽管已经有了C51高级语言,但在实际的科研项目中,采用“C51高级语言”和“汇编语言”混合编程的较为多见。我们认为,在教学中,目前,还是不应忽略汇编语言程序设计方法的介绍,因为在许多实时控制时序和时间要求十分苛刻的场合,仍然非用汇编语言不可;尤其在控制接口硬件时,用汇编语言进行程序设计显得非常简洁;另外,学习汇编语言程序设计,能较好地理解单片机的工作原理,所以学生能用汇编语言设计出高效的程序是一大基本功。绝不能因为有了C51高级语言,就完全废弃汇编语言。

在学习“单片机原理与应用”课程时,学生一般先行开设了“C程序设计”课程,这对学习C51高级语言程序设计有很好的支持作用,因此,本书在“经典内容”的第4章,即单片机程序设计基础这一章中,以一小节的篇幅介绍“C51高级语言程序设计”,并且在本书的第11~13章采用了C51高级语言程序设计的方法,有时还采用了C51高级语言程序设计与汇编语言程序设计相结合的混合编程方法,这种安排对读者而言,应该说具有很好的指导作用。

(6) 介绍基于嵌入式实时操作系统的单片机程序设计方法

在本书第11章,介绍适用于MCS-51系列8位单片机的嵌入式实时操作系统RTX51及其应用方法。该内容有别于“嵌入式系统”课程的内容:一方面,实时操作系统RTX51是适用于MCS-51系列8位单片机的,而“嵌入式系统”课程中所介绍的实时操作系统是适用于32位嵌入式微处理器或64位嵌入式微控制器的;另一方面,实时操作系统RTX51易学好用,笔者在实际的工程项目中使用了它,觉得很好,所以拿出来与读者们共享而放在本书中,且这正是体现本书特色的一个亮点。

(7) 简单介绍单片机应用系统设计的开发环境

本书第10章,主要介绍单片机开发小工具、开发环境Keil μ Vision2和Proteus、以及开发步骤,目的是让读者了解单片机应用系统设计开发的方法和工程设计步骤,建立起单片机应用系统开发的

全局观念。一般说来,单片机开发环境的介绍需要较大的篇幅,并且已有专门的书籍进行了介绍,而要真正掌握单片机的开发环境,只有经过大量的训练方可实现。因此,对于这部分内容本书只做简单介绍,读者必须通过实际操作,不断积累经验,直至熟练运用单片机开发环境。

(8) 与工程应用相结合, 选取完整的设计实例

本书第12~13章,是两个完整的单片机应用系统的设计实例,取材于全国大学生电子设计竞赛的国家级获奖作品,以及取材于实际的工程设计。这两章都给出了完整的系统设计过程,不仅给出了完整的系统硬件电路原理图,而且给出了完整的系统软件设计源程序代码,系统应用程序采用C51高级语言进行编写,或者采用C51高级语言和MCS-51汇编语言混合编程,并且这两章的编写体例是按照电气信息类专业本科毕业设计的论文格式要求进行撰写的,在书中还给出了所设计与制作实物的数码照片,以便增加实际效果。因此,本书的这两章不仅可供从事单片机应用系统开发的工程技术人员参考,还可作为各类电子设计竞赛的培训内容,以及单片机课程设计的参考内容和电气信息类专业学生毕业设计的参考内容。

本书以89S51为典型机,主要论述单片机的基本结构与工作原理,以及单片机应用系统的设计与开发方法。全书内容分为13章:概述;单片机的结构和工作原理;指令系统;单片机程序设计基础;中断系统;定时器/计数器;串行口UART;单片机的常用接口技术;串行总线技术;单片机应用系统的开发环境;基于嵌入式实时操作系统的单片机程序设计方法;基于RTX51的乐曲编辑器和发生器的设计;数控电流源的设计。书后附录给出了18个单片机课程设计课题,以及AT89S51的指令系统。

本书由欧伟明教授担任主编,何静、凌云、刘剑担任副主编。欧伟明教授撰写第1章、第4章、第11~13章,何静博士撰写第5~6章,凌云教授撰写第9章,贺素良教授撰写第3章,刘剑副教授撰写第10章,贺正芸老师撰写第7~8章,龙晓薇老师撰写第2章,欧伟明、凌云、周玉老师撰写附录A,李朝仑老师编写附录B。全书由欧伟明教授统稿和定稿。

本书由肖伸平教授主审。肖伸平教授认真审阅了本书的编写提纲和全部书稿,提出了许多宝贵意见。在本书的撰写过程中,得到了湖南工业大学的张昌凡教授、朱晓青教授、龙永红教授、张满生教授、李祥飞博士和李燕林老师的大力支持,他们给予了作者鼓励和关于本教材的编写意见;此外,还得到了湖南工业大学电气与信息工程学院的毕业生刘张胜、庄永军、楚瑞玉、任杰、周韬、李军杰、蒙毓李、杨敬力、柳红新、卢志波、付贵勇、欧阳文彦和郭仁等的支持,他们对书中部分硬件电路和部分程序的初步调试进行了有益的工作。在此一并表示衷心的感谢!

为了提高读者的动手能力,我们特意研制了多种规格的单片机实验开发板,在学习单片机技术的过程中,如果拥有单片机实验开发板则具有事半功倍的效果;此外,我们还精心制作了与本书相配套的PPT电子课件,它可免费提供给选用本书作为教材的教师。需要者可以通过电子邮件wei13973336675@126.com与作者联系。

欧伟明
2009年7月
于湖南工业大学

参考文献

- [1] 何立民. 单片机高级教程[M]. 北京: 北京航空航天大学出版社, 2000.
- [2] 张毅刚, 彭喜元. 单片机原理与应用设计[M]. 北京: 电子工业出版社, 2008.
- [3] 牛昱光, 等. 单片机原理与接口技术[M]. 北京: 电子工业出版社, 2009.
- [4] 张齐, 朱宁西. 单片机应用系统设计技术[M]. 北京: 电子工业出版社, 2009.
- [5] 欧伟明. 基于MCU、FPGA、RTOS的电子系统设计方法与实例[M]. 北京: 北京航空航天大学出版社, 2007.
- [6] 欧伟明, 等. 电子信息系统设计[M]. 西安: 西安电子科技大学出版社, 2005.
- [7] 姜志海, 赵艳雷. 单片机的C语言程序设计与应用[M]. 北京: 电子工业出版社, 2008.
- [8] 马忠梅, 等. 单片机的C语言应用程序设计[M]. 北京: 北京航空航天大学出版社, 2003.
- [9] 凌玉华, 等. 单片机原理与应用系统设计[M]. 长沙: 中南大学出版社, 2006.
- [10] 周润景, 张丽娜. 基于PROTEUS的电路及单片机系统设计与仿真[M]. 北京: 北京航空航天大学出版社, 2006.
- [11] 徐爱钧, 彭秀华. Keil Cx51 V7.0单片机高级语言编程与 μ Vision2应用实践[M]. 北京: 电子工业出版社, 2005.
- [12] 罗耀华, 蒋志坚. 电子测量仪器原理及应用(II)智能仪器[M]. 北京: 哈尔滨工程大学出版社, 2002.
- [13] 晨风. 嵌入式实时多任务软件开发基础[M]. 北京: 清华大学出版社, 2004.
- [14] 陆应华, 王照平, 王理. 电子系统设计教程[M]. 北京: 国防工业出版社, 2005.
- [15] 杨志亮. Protel 99 SE 电路原理图设计技术[M]. 西安: 西北工业大学出版社, 2002.
- [16] 谢淑如, 郑光钦, 杨渝生. Protel PCB 99 SE电路板设计[M]. 北京: 清华大学出版社, 2001.
- [17] 欧伟明. 嵌入式应用软件任务划分原则的研究[J]. 单片机与嵌入式系统应用, 2007, 6.
- [18] KEIL Software Inc. RTX51 Real-time Kernel[OL]. <http://www.keil.com/rtx51>, 2004.

目 录

第1章 概述	1	2.3.1 89S51的基本组成	24
1.1 单片机的概念与发展过程	1	2.3.2 89S51的CPU	25
1.1.1 单片机的概念	1	2.4 89S51单片机的存储器	27
1.1.2 单片机技术发展过程	1	2.4.1 程序存储器	28
1.1.3 单片机技术发展方向	3	2.4.2 数据存储器	29
1.1.4 常用数制与编码	4	2.5 89S51单片机的时钟电路与时序	31
1.2 单片机的应用领域与嵌入式系统	6	2.5.1 时钟电路	31
1.2.1 单片机的应用领域	6	2.5.2 基本时序单位	32
1.2.2 嵌入式系统的概念	7	2.6 89S51单片机的工作方式	33
1.3 单片机应用系统开发过程简述	8	2.6.1 复位工作方式和复位电路	33
1.3.1 单片机的编程语言	8	2.6.2 低功耗工作方式	35
1.3.2 单片机应用系统的结构	9	2.6.3 ISP编程工作方式	35
1.3.3 单片机的应用模式	10	2.7 89S51单片机的输入/输出接口	36
1.3.4 单片机应用系统开发过程简介	11	2.7.1 P0端口的结构及工作原理	36
1.4 本书特点与教学安排	13	2.7.2 P1端口的结构及工作原理	37
1.4.1 本书编写的指导思想	13	2.7.3 P2端口的结构及工作原理	38
1.4.2 本书特点	16	2.7.4 P3端口的结构及工作原理	38
1.4.3 教学安排建议	16	2.8 本章小结	39
1.5 本章小结	18	2.9 思考题与习题	40
1.6 思考题与习题	19	第3章 指令系统	41
第2章 单片机的结构和工作原理	20	3.1 MCS-51单片机指令概述	41
2.1 MCS-51系列单片机概述	20	3.1.1 指令格式	41
2.2 89S51单片机的引脚功能说明	21	3.1.2 符号说明	42
2.2.1 89S51的引脚图与封装	21	3.2 寻址方式	43
2.2.2 89S51的引脚功能说明	22	3.2.1 寄存器寻址方式	43
2.2.3 89S51的引脚应用特性	23	3.2.2 直接寻址方式	43
2.3 89S51单片机的内部结构	24	3.2.3 寄存器间接寻址方式	43

3.2.4	立即寻址方式	44	4.5.6	C51 函数	81
3.2.5	变址寻址方式	44	4.5.7	C51 语言与汇编语言混合编程 的方法	84
3.2.6	相对寻址方式	44	4.5.8	编写 C51 语言应用程序的基本原则 ...	85
3.2.7	位寻址方式	45	4.5.9	C51 高级语言程序设计举例	87
3.3	89S51 单片机的指令系统	45	4.6	本章小结	89
3.3.1	数据传送类指令	45	4.7	思考题与习题	89
3.3.2	算术运算类指令	47	第 5 章 中断系统	91	
3.3.3	逻辑运算及移位类指令	50	5.1	中断的概念	91
3.3.4	控制转移类指令	52	5.1.1	中断的概念	91
3.3.5	位操作类指令	54	5.1.2	中断的条件与响应过程	92
3.4	本章小结	55	5.2	89S51 的中断系统结构与控制	93
3.5	思考题与习题	56	5.2.1	89S51 的中断源和中断矢量	93
第 4 章 单片机程序设计基础	57	5.2.2	89S51 的中断系统结构	94	
4.1	汇编语言的特点及语句格式	57	5.2.3	中断的控制	95
4.1.1	汇编语言的特点	57	5.3	中断应用举例	100
4.1.2	汇编语言的语句格式	57	5.3.1	单外部中断源系统的设计	100
4.2	汇编语言程序的基本结构	58	5.3.2	多外部中断源系统的设计	101
4.2.1	顺序结构	58	5.4	本章小结	102
4.2.2	分支结构	59	5.5	思考题与习题	102
4.2.3	循环结构	59	第 6 章 定时器 / 计数器	104	
4.3	汇编语言的伪指令与汇编	60	6.1	定时器 / 计数器的结构与控制	104
4.3.1	汇编语言的伪指令	60	6.1.1	89S51 定时器 / 计数器的结构	104
4.3.2	汇编语言的汇编	62	6.1.2	定时器 / 计数器的控制	105
4.4	汇编语言程序设计举例	62	6.2	定时器 / 计数器的 4 种工作方式	106
4.4.1	算术运算程序	62	6.2.1	工作方式 1	106
4.4.2	数制转换程序	66	6.2.2	工作方式 2	107
4.4.3	定时程序	67	6.2.3	工作方式 3	108
4.4.4	查表程序	69	6.2.4	工作方式 0	110
4.4.5	数据极值查找程序	69	6.3	定时器 / 计数器的应用举例	110
4.5	C51 高级语言程序设计	70	6.3.1	脉冲信号的产生	111
4.5.1	C51 的标识符与关键字	70	6.3.2	脉冲宽度的测量	112
4.5.2	C51 语言的数据类型	72			
4.5.3	C51 变量的存储种类和存储器类型 ...	73			
4.5.4	C51 的运算符和表达式	75			
4.5.5	C51 语言的基本语句	77			

6.4 本章小结	112	8.4.2 MC14433 与单片机的接口	150
6.5 思考题与习题	113	8.5 外部并行三总线接口	153
第7章 串行口 UART	114	8.6 大功率器件驱动接口	155
7.1 串行口 UART 的结构与控制	114	8.6.1 光耦接口	155
7.1.1 串行口的结构	114	8.6.2 继电器接口	156
7.1.2 串行口的控制	115	8.6.3 双向晶闸管输出接口	157
7.2 串行口的工作方式	116	8.6.4 固态继电器输出接口	157
7.2.1 工作方式 0	116	8.7 本章小结	158
7.2.2 工作方式 1	117	8.8 思考题与习题	159
7.2.3 工作方式 2	118	第9章 串行总线技术	160
7.2.4 工作方式 3	119	9.1 串行通信基本知识	160
7.3 串行口应用举例	119	9.1.1 并行通信和串行通信	160
7.3.1 串行口 UART 扩展并行 I/O 接口	119	9.1.2 串行通信的传输方式	160
7.3.2 串行口 UART 实现双机异步 串行通信	121	9.1.3 通信协议	161
7.3.3 串行口 UART 实现多机异步 串行通信	122	9.2 串行通信 EIA 系列总线标准及其接口 ...	164
7.4 本章小结	127	9.2.1 RS-232C 总线	164
7.5 思考题与习题	128	9.2.2 RS-422/485 总线	165
第8章 单片机的常用接口技术	129	9.2.3 单片机与 PC 之间的通信	167
8.1 键盘接口	129	9.3 三线制同步串行总线接口	169
8.1.1 独立式按键	130	9.3.1 SPI 总线简介	169
8.1.2 矩阵键盘	131	9.3.2 MICROWIRE 接口简介	171
8.2 LED 显示器接口	135	9.3.3 SPI 接口单片机 P89LPC93x 与 ADC 器件 AD7810 的通信	171
8.2.1 LED 数码管	136	9.3.4 MICROWIRE 接口 E ² PROM 器件 AT93C46 的应用	174
8.2.2 LED 数码管的静态显示接口	137	9.4 I ² C 总线	177
8.2.3 LED 数码管的动态显示接口	138	9.4.1 I ² C 总线简介	177
8.3 DAC 接口	140	9.4.2 P89C66x 系列单片机 I ² C 总线 编程规范	178
8.3.1 DAC0832 与单片机的接口	140	9.4.3 I ² C 接口 DAC 转换器 MAX517 的应用	181
8.3.2 TLC5618 与单片机的接口	143	9.5 CAN 总线	183
8.4 ADC 接口	146	9.5.1 CAN 总线简介	184
8.4.1 ADC0809 与单片机的接口	147		

9.5.2 CAN 总线控制器	185	11.1.1 嵌入式系统的特征	225
9.5.3 CAN 总线通信接口设计实例	185	11.1.2 嵌入式实时操作系统的概念	226
9.6 USB 总线	187	11.2 在电子系统设计中引入实时操作系统的意义	226
9.6.1 USB 总线原理	187	11.2.1 两种软件开发模式的比较	227
9.6.2 USB 总线通信接口设计实例	189	11.2.2 嵌入式应用中使用 ERTOS 的必要性	227
9.7 1-Wire 单总线	191	11.2.3 嵌入式操作系统环境下的应用软件设计	228
9.7.1 1-Wire 单总线简介	191	11.2.4 嵌入式操作系统环境下的应用软件调试	229
9.7.2 单总线温度传感器 DS18B20 及其应用	192	11.3 嵌入式实时操作系统 RTX51 介绍 ...	229
9.8 本章小结	196	11.3.1 RTX51 的技术参数	229
9.9 思考题与习题	197	11.3.2 几个概念	230
第 10 章 单片机应用系统的开发环境	198	11.3.3 RTX51 Tiny 内核分析	233
10.1 单片机应用系统的开发方法	198	11.3.4 RTX51 Tiny 内核源代码	236
10.1.1 单片机应用系统的开发过程	198	11.4 基于 RTX51 的单片机程序设计方法 ...	238
10.1.2 单片机应用系统的仿真调试	201	11.4.1 目标系统需求	238
10.2 Keil μ Vision2 集成开发环境	204	11.4.2 软件设计指导方针	238
10.2.1 Keil μ Vision2 的主要特性	205	11.4.3 任务划分的原则	239
10.2.2 Keil μ Vision2 的使用方法	206	11.4.4 应用程序的架构	241
10.3 Proteus 仿真软件	215	11.5 本章小结	243
10.3.1 Proteus 主界面介绍	215	11.6 思考题与习题	243
10.3.2 Proteus 绘制电路原理图	217	第 12 章 基于 RTX51 的乐曲编辑器	
10.3.3 Proteus 与 Keil μ Vision2 的联调 ...	219	和发生器的设计	244
10.4 单片机应用系统开发小工具	220	12.1 设计任务	244
10.4.1 波特率初值计算工具	221	12.2 方案设计与论证	244
10.4.2 数码管编码器	221	12.2.1 以 FPGA 为核心的实现方案	244
10.4.3 定时器计算工具	222	12.2.2 以 MCU 为核心的实现方案	245
10.4.4 串口调试助手	222	12.3 系统硬件设计	245
10.5 本章小结	223	12.3.1 系统硬件电路原理图	245
10.6 思考题与习题	223	12.3.2 人机交互界面	246
第 11 章 基于嵌入式实时操作系统的单片机程序设计方法	225	12.4 基于 RTX51 的系统软件设计	247
11.1 嵌入式实时操作系统的概念	225	12.4.1 乐曲的表示方法	247

12.4.2	编辑乐曲的软件实现方法	249	13.4.2	D/A 转换模块、恒流源 模块的 电路图	274
12.4.3	播放乐曲的软件实现方法	250	13.4.3	数据采集模块的电路图	276
12.4.4	系统软件流程框图	251	13.4.4	辅助电源、主电源的电路图	277
12.5	系统源程序清单	252	13.5	系统软件设计	277
12.5.1	C51 高级语言主程序	252	13.5.1	主程序流程框图	278
12.5.2	读 AT24C02 汇编语言子程序	261	13.5.2	设置输出电流给定值程序 流程框图	278
12.5.3	写 AT24C02 汇编语言子程序	263	13.5.3	设置电流步进值程序流程框图	279
12.5.4	键盘扫描汇编语言子程序	265	13.5.4	键盘扫描程序流程框图	280
12.5.5	实时操作系统 RTX51 Tiny 内核程序	266	13.6	系统测试方法与结果分析	280
12.6	本章小结	266	13.6.1	测试使用的仪器	280
第 13 章	数控电流源的设计	267	13.6.2	恒流特性的测试	281
13.1	设计任务	267	13.6.3	电流步进值为 1 mA 的测试	281
13.2	方案设计与论证	267	13.6.4	纹波电流的测试	282
13.2.1	D/A 转换模块设计方案的 论证与比较	268	13.6.5	输出电流范围的测试	282
13.2.2	恒流源模块设计方案的 论证与比较	268	13.6.6	输出电压的测试	282
13.2.3	数据采集模块设计方案的 论证与比较	269	13.6.7	1~99 mA 内任意电流步进值 设置功能的测试	283
13.2.4	辅助电源、主电源设计方案的 论证与比较	270	13.6.8	测试结果分析	283
13.2.5	键盘、显示器设计方案的 论证与比较	271	13.7	系统使用说明书	284
13.3	理论计算与 EWB 仿真	272	13.7.1	键盘界面	284
13.3.1	采样电阻值的确定	272	13.7.2	菜单操作	284
13.3.2	D/A 转换器分辨率的确定	272	13.8	系统源程序清单	285
13.3.3	TLC5618 参考电压的确定	272	13.8.1	C51 高级语言主程序	285
13.3.4	主电源参数的确定	272	13.8.2	键盘扫描汇编语言子程序	294
13.3.5	用 EWB 进行电路仿真	273	13.8.3	写 TLC5618 的汇编语言子程序 ...	295
13.4	系统硬件设计	273	13.8.4	读 MC14433 的汇编语言子程序 ...	296
13.4.1	MCU 微控制器、键盘、显示器 电路图	274	13.8.5	显示缓冲器的汇编语言子程序	297
			13.9	本章小结	298
			附录 A	单片机课程设计	299
			附录 B	89S51 指令表	308
			参考文献	312

第1章 概 述

本章主要对单片机的概念与发展过程、单片机的应用领域与嵌入式系统的概念、单片机应用系统开发过程、本书编写指导思想与教学安排建议等内容,进行简要介绍,以便读者对单片机及其应用技术有初步的了解。

1.1 单片机的概念与发展过程

自1971年微型计算机问世以来,由于实际应用的需要,微型计算机向着两个方向发展:一个是向着高速度、大容量、高性能的高档微机方向发展;而另一个则是向着稳定可靠、体积小、功耗低、价格低廉的单片机方向发展。单片机是微型计算机的一个重要分支,它的出现是计算机技术发展史上的一个重要里程碑,它使计算机从海量存储与高速复杂数值计算进入到智能化控制领域。从此,计算机技术的两个重要领域——通用计算机领域和嵌入式计算机领域都获得了极其重大的进展。

1.1.1 单片机的概念

单片机在一块半导体硅片上集成了计算机的所有基本功能部件,包括中央处理器(CPU)、存储器(RAM和ROM)、输入/输出接口电路、中断系统、定时器/计数器和串行通信接口电路等。因此,单片机只需要与适当的软件及适当的外部设备相结合,就可以构成一个完整的计算机应用系统。

单片微型计算机(Single Chip MicroComputer, SCMC)简称单片机。“单片机”一词,真实地反映了单片机的形态和本质。由于单片机主要应用于各种测控系统中,所以目前国外已普遍地将单片机称为微控制器(MicroController Unit, MCU)。鉴于单片机在应用时通常处于测控系统的核心地位并融入被控对象中,即以嵌入被控对象的方式而进行使用,为了强调其嵌入式应用特点,又称之为嵌入式控制器(Embedded MicroController Unit, EMCU)。

在汉语中,单片机的称呼简洁通俗,以至于单片机一词已约定俗成,仍沿用至今,所以本书仍使用“单片机”一词。

1.1.2 单片机技术发展过程

单片机诞生于20世纪70年代,美国Intel公司自1976年宣布并于1977年推出MCS-48单片机以来,世界上的单片机技术已经走过了30余年的历程。我国自20世纪70年代末80年代初就开始进行单片机的应用与开发工作,1987年10月27日,在上海成立中国微计算机单片机学会,中国的单片机开发与应用经历了20余年。几十年来,单片机不是以其位数的高低来决定单片机的先进性的,而是以如何适合千变万化的应用产品的需求、高性能价格比的配置来决定单片机的优劣。因此,高性能价格比、多功能、低功耗的8位单片机一直是单片机的主角。

单片机作为微型计算机的一个重要分支,应用面很广,发展很快。如果将8位单片机的推出作为起点,那么单片机的发展历史大致可分为以下几个阶段:

1. 第一阶段 (1976—1978): 单片机探索阶段

工控领域对计算机提出了嵌入式应用的要求,首先是实现单芯片形态的计算机,以满足构成大量中小型智能化测控系统的要求。因此,这一阶段的任务是探索计算机的单芯片集成。“单片机”一词即由此而来。

在单片机探索阶段,单片机体系结构有两种模式,即通用CPU模式和专用CPU模式。

- (1) 通用CPU模式。采用通用CPU和通用外围单元电路的集成方式,这种模式以美国Motorola公司的MC6801单片机为代表,它将通用CPU、增强型的6800+和6875(时钟)、6810(128 B RAM)、 2×6830 (1 KB ROM)、 $1/2 6821$ (并行I/O端口)、 $1/3 6840$ (定时器/计数器)、6850(串行I/O端口)集成在一个芯片上构成,使用6800CPU指令系统。
- (2) 专用CPU模式。采用专门为嵌入式系统要求设计的CPU与外围电路集成的方式。这种模式以美国Intel公司的MCS-48单片机为代表,其CPU、存储器、定时器/计数器、中断系统、I/O端口、时钟以及指令系统都是按嵌入式系统要求而专门设计的。

事实证明,这两种模式都是可行的。专用CPU模式能充分满足嵌入式应用的要求,成为今后单片机发展的主要体系结构模式;通用CPU模式则与通用计算机兼容,应用系统开发方便,成为后来嵌入式微处理器的发展模式。

2. 第二阶段 (1978—1982): 单片机完善阶段

Intel公司在MCS-48单片机基础上推出了完善的、典型的MCS-51系列单片机。它在以下几个方面奠定了典型的通用总线型单片机体系结构。

- (1) 规范的总线结构。有8位数据总线、16位地址总线以及多功能的异步串行通信接口UART。
- (2) CPU外围功能单元电路的集中管理模式。
- (3) 提供体现工控特性的位地址空间及位操作方式。
- (4) 指令系统趋于丰富和完善,并且增加了许多突出控制功能的指令。

3. 第三阶段 (1982—1990): 微控制器形成阶段

8位单片机的巩固和发展以及16位单片机的推出阶段,正是单片机向微控制器发展的阶段。Intel公司推出的MCS-96系列单片机,将一些用于测控系统的模/数转换器、程序运行监视器、脉宽调制器等纳入芯片内部,体现了单片机的微控制器特征。随着MCS-51系列的广泛应用,许多半导体制造厂商竞相使用80C51为内核,将许多测控系统中使用的电路技术、接口技术、多通道A/D转换部件、可靠性技术等应用到单片机中,增强了外围电路的功能,强化了智能控制的特征。

为了满足测控系统的嵌入式应用要求,这一阶段单片机的主要技术发展方向是满足测控对象要求的外围电路的增强,从而形成了不同于Single Chip MicroComputer的特点的微控制器,微控制器MCU(MicroController Unit)一词缘于这一阶段。至今,微控制器MCU是国际上对单片机的标准称呼。

在微控制器形成阶段,单片机技术发展的主要方面如下所示。

- (1) 外围功能电路集成到芯片内部。满足模拟量输入的ADC;满足伺服驱动的PWM;满足高速I/O控制的高速I/O端口;保证程序可靠运行的程序监视定时器WDT。
- (2) 出现了为满足串行外围扩展要求的串行扩展总线及接口,如SPI、I²C、MicroWire、USB和1-Wire等。
- (3) 出现了为满足分布式系统、突出控制功能的现场总线接口,如CAN总线。

(4) 在程序存储器方面,则迅速推出 OTP (一次可编程) 单片机,为单片机的单片应用创造了良好的条件,随后 Flash ROM 单片机的推广,为最终取消单片机外部程序存储器奠定了良好的基础。

4. 第四阶段 (1990 年至今): 微控制器全面发展阶段

随着单片机在各个领域全面深入地发展和应用,出现了高速、大寻址范围、强运算能力的 8 位/16 位/32 位通用型单片机,以及小型廉价的专用型单片机。其特点是,百花齐放,技术创新,以满足日益增长的广泛需求,将单片机用户带入了一个可以广泛选择的时代。

1.1.3 单片机技术发展方向

单片机的发展趋势是向 CMOS 化、低功耗、小体积、大容量、高性能、低价格和外围电路内装化等几个方面发展。

1. 主流机型发展趋势

在未来较长一段时间内,8 位单片机将仍是主流机型,许多厂家还会不断改进与完善 8 位单片机,使 8 位单片机不断保持其活力;在满足高速数字处理方面,32 位单片机会发挥重要作用;16 位单片机的空间有可能被 8 位单片机或 32 位单片机挤占。

2. CMOS 化趋势

CMOS 工艺很早就已出现,它具有十分优异的性能,只是运行速度较慢,因而长期被冷落。HCMOS 工艺出现后,HCMOS 器件得到了飞速的发展。从第三代单片机起,开始淘汰非 CMOS 工艺。CMOS 芯片除了低功耗特性之外,还具有功耗的可控性,使单片机可以工作在功耗精细管理状态。另外,如今的数字逻辑集成电路、计算机外围器件都已普遍 CMOS 化。

3. RISC 体系结构单片机的大发展

早期的单片机大多是复杂指令集计算机 CISC 体系结构,指令复杂,指令代码、周期数不统一;指令运行很难实现流水线操作,大大阻碍了运行速度的提高。例如,MCS-51 系列单片机,晶振频率为 12 MHz 时,单周期指令速度仅 1 MIPS。虽然单片机对运行速度的要求远不如通用计算机或数字信号处理器 (DSP) 对指令运行速度的要求,但速度的提高会带来许多好处,并拓宽单片机应用领域。如果采用精简指令集计算机 RISC 体系结构,精简指令后绝大部分成为单周期指令,而且通过增加程序存储器的宽度 (例如从 8 位增加到 10 位、12 位、14 位),实现一个地址单元存放一条指令。在这样的体系结构中,很容易实现并行流水线操作,其结果是能大大提高指令运行速度,在相同的运行速度要求下,可大大降低晶振频率,这有利于获得良好的电磁兼容性效果。

4. 大力发展专用型单片机

专用单片机是专门针对某一类产品系统要求而设计的。使用专用单片机可最大限度地简化系统结构,资源利用效率最高。在大批量使用时,有可观的经济效益和可靠性效益。

5. 推行串行扩展总线

目前,计算机外围器件技术发展的一个重要方面是串行接口技术的发展。采用串行接口,可以大大减少器件的引脚数量,简化系统结构。采用串行接口虽然较之并行接口数据传输速度慢,但由于串行传输速度的不断提高,加之单片机所面对被控对象的有限速度要求,使单片机应用系统中的串行扩展技术有了很大发展,而单片机的并行接口技术则日渐衰退。目前,许多原有带并行总线的单片机系列,推出了许多删去并行总线的非总线型单片机。

1.1.4 常用数制与编码

计算机是用来处理数字信息的，单片机也是如此。各种数据信息、非数据信息在进入计算机前必须转换成二进制数、二进制编码。

1. 常用数制

每一个数在计算机中是用半导体器件的物理状态来表示的，即利用半导体器件的高电平状态和低电平状态，并且一般用半导体器件的高电平状态表示“1”，用半导体器件的低电平状态表示“0”。不同的0和1组合，就可以表示不同的数或值，例如，10001000（二进制数）就表示十六进制数的88H，即十进制数的136。

计算机在对数进行处理时，只认识0和1，即只认识二进制的数。然而，当数值较大时，二进制的数很难为人们所辨认，并且人们在编写单片机源程序时，书写二进制的数也很不方便。因此，在单片机技术应用中，人们常使用二进制、十进制和十六进制。它们之间的相互关系如表1.1所示。

在使用不同的数制时，各种数制用简码来标示，如十进制数用D（decimal）标示或省略，二进制数用B（binary）来标示，十六进制数用H（hexadecimal）或0x来标示。

例如，十进制数123可以表示为123D或123；二进制数1011可表示为1011B；在编写汇编语言源程序时，十六进制数3A4表示为3A4H，十六进制数A4表示为0A4H；在编写C51高级语言源程序时，十六进制数3A4表示为0x3A4或0x3a4，十六进制数A4表示为0xA4或0xa4。

表 1.1 二进制、十进制和十六进制数的相互关系

十进制数	二进制数	十六进制数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

2. 常用编码

由于计算机只能识别二进制数的0和1，所以计算机处理的任何信息必须以二进制形式表示，这些二进制形式的代码称为二进制编码（Encode）。计算机中常用的二进制编码有BCD码和ASCII码等。

(1) BCD码——二-十进制码

在计算机中，各种数据要转换为二进制代码才能进行处理，而人们习惯于使用十进制数，所以在计算机的输入/输出中仍采用十进制数，这样就产生了用4位二进制数表示1位十进制数的方法，这种用于表示十进制数的二进制代码称为二-十进制代码（Binary Coded Decimal），简称为BCD码。

它具有二进制数的形式以满足计算机的要求，又具有十进制的特点，只有十种有效状态。最常用的BCD码有8421BCD码，如表1.2所示。

表 1.2 8421BCD 码表

十进制数	8421BCD 码	二进制数
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	0001 0000	1010
11	0001 0001	1011
12	0001 0010	1100
13	0001 0011	1101
14	0001 0100	1110
15	0001 0101	1111
16	0001 0110	10000

表1.2中的最后7行所介绍的是用1个字节表示2位十进制数的代码，称为压缩BCD码。而用1个字节表示1位十进制数（高4位补0）的代码，称为非压缩BCD码。在某些情况下，单片机也可以对压缩BCD码表示的数直接进行运算，两个压缩BCD码进行运算时，可能会产生非法码（代码在1010B~1111B范围），此时需要对运算结果进行调整。

另外，BCD码还有余3码、格雷码等，相关内容请参见数字电子技术教材。

(2) ASCII 码

ASCII (American Standard Code for Information Interchange) 码是一种字符编码，是美国信息交换标准代码的简称，如表1.3所示。它由7位二进制数码构成，共表示128个字符，包括英文字母、数字、标点符号、控制字符和一些其他字符。ASCII码用于计算机和计算机之间、计算机和外围设备之间的文字交互。

比如，字母“A”的编码是65，字母“a”的编码是97，PC键盘上的空格键的编码是32，等等。当然，仅用ASCII码不能完全表示所有的字符，如汉字、韩文、日文等都无法用ASCII码直接表示。

表 1.3 ASCII 码字符表

高 位	低 位																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
0	000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	DEL	BS	HT	LF	VT	FF	CR	SO	SI
1	001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	010	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL