

单片机

入门捷径与实例

玉 瑞 贡 岩◎编著



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

单片机入门捷径与实例

玉 瑞 贡 岩 编 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从初学者的角度对他们在学习单片机过程中遇到的问题进行了归纳,选取了 200 多个问题,以“非典型”问答方式介绍 MCS-51 系列及其兼容单片机的基本知识和接口技术。主要内容包括由外而内地介绍 MCS-51 系列及其兼容单片机的基本结构,指令系统及程序设计,C51 语言程序设计,中断系统、定时器/计数器、串行口的工作原理,存储器扩展和接口技术等,并精选择了一些程序设计、中断系统、定时器/计数器应用、串行口应用、接口技术及典型的新型接口器件的应用程序实例。最后给出了部分汇编语言和 C51 语言程序索引,方便读者设计程序时参考。

本书可供工业自动化、电子信息工程、应用电子技术、计算机应用等相关本、专科专业学生结合教材使用,也可供自学考试人员、单片机应用系统开发的工程技术人员参考。

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

单片机入门捷径与实例 / 玉瑞, 贡岩编著. —北京: 电子工业出版社, 2014.6

ISBN 978-7-121-23150-6

I. ①单… II. ①玉… ②贡… III. ①单片微型计算机—基本知识 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2014)第 091896 号

策划编辑: 杨 博

责任编辑: 周宏敏 文字编辑: 张 迪

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: 787×1 092 1/16 印张: 15.5 字数: 396.8 千字

印 次: 2014 年 6 月第 1 次印刷

定 价: 39.80 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。

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

服务热线: (010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

前言

单片机因在控制、机电一体化、智能仪表、消费类电子产品等方面的广泛应用，成为各类高等学校电类和机械类各专业的重要专业课程。MCS-51 及其兼容系列单片机是国内目前应用最广泛的一种 8 位单片机，也是高等学校单片机原理课程教学内容的主体。

单片机原理课程，严格来说，不是一门理论课程，而是一门具有较强实践性的专业技术课程，其主要内容针对某一特定系列的单片机产品。我们所面对的单片机，是一个高度集成化的电路芯片，没有精确的电路模型和电路分析，除引脚外，其硬件方面的内容大多是以寄存器为学习载体的功能部件。这对刚从基础理论课程转入专业课程学习的学生来说，是一种不小的挑战，对学生理解能力、动手能力有较大的考验。他们在学习过程中对基本概念和应用中一些问题的理解，往往似是而非。事实上，初学者在学习过程中，有着各种各样的问题，而一般的教科书或简或繁的叙述方式，难以直接针对初学者所面临的问题。教科书所提供的思考复习题，一般只起到复习巩固知识点的作用，无法解答学生们心中的疑惑。解决初学者面临的这些问题，对于他们学好、用好单片机及接口技术，并举一反三，是非常重要的。

作者在多年的单片机教学过程中，对初学者的这些问题深有体会，针对初学者遇到的问题进行了归纳，选取了 200 多个问题，以问答的方式编写成书，作为学习单片机与接口技术的学习参考书和辅助读物，供读者结合教科书使用。

本书结合初学者对单片机认识的自然过程及教学和一般教科书的编排顺序，第 1 章介绍单片机和微机的基本知识，解决“我所学的单片机是个什么样子”的问题；第 2 章和第 3 章分别从引脚功能、内部存储单元及寄存器入手，由外到里介绍单片机所具有的功能和用途，以及单片机的内部基本功能、程序和数据的存储空间；第 4 章把对指令系统的认识与程序设计相结合，说明指令系统的功能和用法，以及各类符号的作用和使用方法，解决程序阅读中的指令理解和程序设计过程中的指令选用问题，解答学习中可能出现的问题，纠正初学者在指令理解和程序设计中常见的误区；第 5 章在 C 语言的基础上，根据 C51 语言与 C 语言、汇编语言的异同，说明采用 C51 语言的使用特点和进行单片机程序设计时要注意的问题；第 6 章说明中断系统的作用和用法、中断响应的过程和条件、中断优先权和嵌套，以及外中断的程序设计；第 7 章从分析定时器/计数器的结构图出发，说明定时器/计数器的用法、工作方式、相关寄存器所起的作用和计数初值计算，给出了多个查询方式和中断方式的程序设计实例；第 8 章通过串行口的工作条件、结束条件和结束状态三要素，说明串行口的作用和用法，并用完整的查询方式和中断方式程序说明双机通信和多机主从通信的程序设计；第 9 章从单片机引脚的利用和存储器的扩展入手，介绍单片机

基本的接口技术。在介绍存储器扩展方式时,注重对扩展电路的分析,提高读者对扩展存储器使用方法的认识。同时介绍了 I/O 接口的扩展方法;第 10 章和第 11 章结合单片机应用系统构成的实际情况,介绍当前广泛使用的一些接口器件及在单片机系统中的使用方法,这些器件工作过程的理解和使用是很多初学者面临的大难题。第 10 章主要介绍常见的输入和输出器件,首先介绍 LED 数码管的动态显示分析和键盘功能,并从最简单的程序设计入手,逐渐优化程序,以求能把读者的思路引向深入。还以实例介绍了应用较广的 LCD 液晶显示模块、LED 数码管显示驱动器件 MxA7219 的使用方法和程序设计。第 11 章是学习完基本内容后的提高,介绍集成温度传感器 DS18B20、并行接口实时时钟 DS12C887、串行实时时钟 DS1306 的功能、接口技术和程序设计方法。

为便于初学者学习,书中用大量实例说明单片机各功能部件的程序设计及接口器件的控制和通信,所举程序例子的任务尽量简单,以便读者更好地理解 and 掌握程序的主要任务,各指令均结合其在程序中的作用加以注释。书中几乎所有程序设计实例都配有汇编语言和 C51 语言的程序,方便初学者对照学习。书中程序,不管长短,均是完整、可运行的,并均经过实验检验。书末给出了程序索引,方便读者进行程序设计时参考。

本书在编写过程中得到了华南理工大学毛宗源教授的热情鼓励,他审阅了书中的内容并提出宝贵的意见;华南理工大学自动化科学与工程学院的杜启亮老师、研究生肖志远同学给予了很大帮助;广州大学电气工程及其自动化专业 2006 级的黄剑庭同学为本书绘制了多幅电路图。在此一并对他们表示衷心的感谢。

本书适合作为各类高等院校相关专业的教学参考书。对书中的错误和不当之处,敬请读者指正。

编 著 者

目 录

第 1 章 对单片机的基本认识	1
1.1 单片机是什么?	1
1.2 单片机是什么样子?	1
1.3 单片机有哪些型号?	2
1.4 从外形看, 怎么知道单片机的功能?	2
1.5 怎么知道单片机是否在工作?	2
1.6 单片机和微机有什么区别?	2
1.7 单片机与专用集成电路有什么区别?	3
1.8 单片机与可编程控制器 (PLC) 有什么区别?	3
1.9 单片机的组成结构是怎样的?	4
1.10 什么是普林斯顿结构和哈佛结构?	5
1.11 MCS-51 单片机有什么功能?	5
1.12 AT89S 系列单片机的功能有什么特点?	6
1.13 单片机用得多么?	6
第 2 章 由外到里认识单片机——单片机的引脚功能	8
2.1 单片机的引脚有些什么作用?	8
2.2 单片机接晶体振荡器有什么作用?	9
2.3 单片机有哪些时序单位?	9
2.4 $\overline{EA}$ 引脚一定要连接吗?	10
2.5 复位有什么作用?	10
2.6 复位以后单片机处于什么状态?	10
2.7 什么是 I/O 端口?	11
2.8 为什么要外接存储器?	12
2.9 为什么要复用 P0 口?	12
2.10 扩展外部存储器, 一定要用到 P2 口吗?	12
2.11 外部数据存储器的地址资源用不完, 能否实现外借?	13
2.12 并行口的“读引脚”和“读锁存器”操作有何区别?	13
2.13 P1 口的结构和工作是怎样的?	14

2.14	P3 口的结构和工作是怎样的？	15
2.15	P2 口的结构和工作是怎样的？	17
2.16	P0 口的结构和工作是怎样的？	18
2.17	为什么 I/O 口锁存器的位置 1，才能进行输入操作？	20
2.18	I/O 口的输入/输出操作需要先进行初始化吗？	20
2.19	AT89C2051、AT89S2051 等单片机为什么只有 20 个引脚？	20
2.20	怎样才能使单片机工作？	21
2.21	什么是单片机最小系统？	22
2.22	AT89S 系列单片机有什么新的引脚功能？	22
第 3 章 由外到里认识单片机——单片机的内部基本功能		23
3.1	单片机的 CPU 在哪里？	23
3.2	程序状态字 PSW 起什么作用？	23
3.3	程序计数器 PC 起什么作用？	24
3.4	特殊功能寄存器有什么作用？	25
3.5	什么是程序存储器地址空间？	26
3.6	未使用的程序存储器单元里的内容是什么？	27
3.7	什么是数据存储器地址空间？	27
3.8	8052 等器件有 256 字节内部数据存储器单元，是否会与特殊功能寄存器产生地址冲突？	29
3.9	如何选择使用工作寄存器组？	29
3.10	AT89S 系列单片机有哪些新的特殊功能寄存器？	29
3.11	什么是 AT89S 系列单片机的 ISP 系统内编程？	32
第 4 章 指令系统和程序设计		36
4.1	指令跟硬件功能有关系吗？	36
4.2	还需要人工编译汇编语言源程序吗？	36
4.3	理解机器码有什么作用，有必要吗？	36
4.4	字节型传送指令 MOV 有什么特点？	37
4.5	把累加器写成 A 与写成 ACC 有什么不同？	37
4.6	指令格式中的 direct 是什么意思？	38
4.7	一个内部数据存储器单元应该用 direct 还是 Rn 形式表示好？	38
4.8	如何理解指令中的“@”标识符？	39
4.9	如何理解指令中的“#”标识符？	39
4.10	如何区分指令中的数值是地址还是常数？	39
4.11	什么是堆栈操作？	40
4.12	堆栈操作指令的特点是什么？	40
4.13	工作寄存器 Rn 和累加器 A 可以堆栈吗？	41
4.14	堆栈区的起点可以自由设定吗？	41
4.15	堆栈区的大小是固定的吗？	41

4.16	如何实现数据交换？	42
4.17	传送指令 MOV _C 有什么特点？	42
4.18	传送指令 MOV _X 有什么特点？	43
4.19	算术运算指令的特点是什么？	44
4.20	可以实现累加器 A 的内容自己相加吗？	45
4.21	如何实现 DPTR 的内容减 1？	45
4.22	乘法指令和除法指令的操作有什么特点？	46
4.23	有哪些针对累加器 A 本身的逻辑操作指令？	46
4.24	如何进行与、或、异或运算？	47
4.25	LJMP 和 AJMP 指令的区别是什么，如何选用？	48
4.26	LCALL 和 ACALL 指令的区别是什么，如何选用？	48
4.27	子程序是以 LCALL、ACALL 指令开始的吗？	49
4.28	有几种子程序返回指令，其用途是什么？	49
4.29	“SJMP rel” 和 “JMP @A+DPTR” 指令有什么作用？	49
4.30	“JZ rel” 和 “JNZ rel” 指令是对哪个寄存器的？	50
4.31	CJNE 指令的功能是什么？	51
4.32	DJNZ 指令在程序中的作用是什么？	51
4.33	如何理解转移指令中的 rel？	51
4.34	空操作 NOP 指令是没有用的指令吗？	52
4.35	如何进行可寻址位的数据传送？	53
4.36	可寻址位有哪些运算指令？	53
4.37	可寻址位有哪些转移指令？	54
4.38	在程序中如何表示一个可寻址的位？	54
4.39	如何区分一条指令是字节操作指令还是位操作指令？	54
4.40	子程序有什么作用，在什么情况下要采用子程序？	55
4.41	设计子程序有什么需要注意的地方？	55
4.42	什么是保护和恢复现场？	56
4.43	子程序的调用可以嵌套吗？	56
4.44	为什么中断响应不自动保护“现场”？	57
4.45	主程序的主体应放在程序存储器的哪一部分？	57
4.46	什么是标号？	57
4.47	“\$” 是什么符号？	58
4.48	使用伪指令有什么好处？	58
4.49	有哪些常用的伪指令？	58
4.50	单片机的程序结构与 C 语言一样吗？	61
4.51	在什么情况下能用 “DA A” 指令？	61
4.52	如何进行压缩 BCD 码数的减法运算？	62
4.53	在汇编语言源程序中，如何表示 BCD 码常数？	63
4.54	如何理解和编写软件延时子程序？	63
4.55	指令有“常用”和“不常用”之分吗？	64

第 5 章 单片机的 C51 语言程序设计	65
5.1 C51 语言和 C 语言的程序有什么异同?	65
5.2 C51 语言程序中变量的存储器类型指的是单片机本身的存储器吗?	65
5.3 C51 语言如何对程序中的变量确定存储单元?	66
5.4 C51 语言如何定义和访问单片机的特殊功能寄存器?	66
5.5 C51 语言程序如何直接访问单片机内部数据存储器单元?	67
5.6 C51 语言如何定义和访问单片机的可寻址位?	67
5.7 C51 语言程序如何访问单片机外部数据存储器和 I/O 接口器件?	68
5.8 C51 语言的逻辑运算符与汇编语言中的逻辑运算指令是相同的吗?	69
5.9 C51 语言的位运算符与汇编语言中的位操作指令是相同的吗?	69
5.10 C51 语言的指针和汇编语言的间接寻址方式有什么区别?	69
5.11 在 C51 语言程序中如何使用单片机的中断功能?	70
5.12 C51 语言程序的中断函数会自动保护现场吗?	71
5.13 在 C51 语言程序中如何使用定时器/计数器?	71
5.14 在 C51 语言程序中如何使用工作寄存器?	71
5.15 C51 语言程序可以和汇编语言程序放在一起吗?	71
5.16 如何提高 C51 语言程序的执行速度?	72
5.17 采用 C51 语言设计的程序, 是从 main() 开始执行的吗?	73
第 6 章 中断系统应用	74
6.1 为什么要中断?	74
6.2 中断系统有什么作用?	74
6.3 中断从哪里来?	75
6.4 如何知道中断来了?	75
6.5 中断了, 干什么?	75
6.6 中断请求什么时候来?	76
6.7 中断服务子程序在哪里?	76
6.8 中断完了干什么?	76
6.9 有多个中断源同时申请中断, 怎么办?	77
6.10 有一个中断正在处理(响应)时, 又有新的中断申请, 怎么办?	77
6.11 中断系统应具有什么功能?	77
6.12 MCS-51 单片机有哪几个中断源?	78
6.13 对外部中断信号有什么要求?	79
6.14 在中断响应后, 进入中断服务前要满足什么条件?	79
6.15 从中断申请到进入程序需要多长时间?	80
6.16 响应中断后, 如何处理中断申请标志?	80
6.17 如何设置中断系统?	81
6.18 能否用程序控制对中断申请的取舍?	82
6.19 “AJMP \$” 和 “SJMP \$” 是“等待中断”的指令吗?	82
6.20 可以用转移指令退出中断服务程序吗?	83

6.21	如何掌握中断优先级的顺序？	83
6.22	在中断服务程序中如何利用工作寄存器？	83
6.23	外中断应用程序举例	84
第 7 章 定时器/计数器应用		88
7.1	为什么需要定时器/计数器？	88
7.2	定时器/计数器的工作原理是怎样的？	88
7.3	定时器/计数器有几种工作方式？	89
7.4	如何设置定时器/计数器的工作？	89
7.5	定时器/计数器在方式 0 是如何工作的？	91
7.6	定时器/计数器在方式 1 是如何工作的？	91
7.7	定时器/计数器在方式 2 是如何工作的？	92
7.8	定时器/计数器 0 在方式 3 是如何工作的？	93
7.9	如何使定时器/计数器按我们的希望溢出？	94
7.10	如何计算定时器/计数器的计数初值？	94
7.11	如何选用定时器/计数器的工作方式？	95
7.12	如何实现超过定时器/计数器范围的定时？	96
7.13	单片机上电或复位后，定时器/计数器会不会自动开始工作？	96
7.14	用方式 0、1、3 进行周期性定时/计数时，为什么要重新装入计数初值？	96
7.15	如何设计定时器/计数器的应用程序？	96
7.16	定时器/计数器应用举例	97
第 8 章 串行口应用		115
8.1	什么是同步和异步串行通信？	115
8.2	什么是单工、半双工、全双工方式？	115
8.3	实现串行通信要用哪些引脚？	116
8.4	为什么需要时钟控制串行通信？	116
8.5	MCS-51 单片机的串行口是哪钟接口协议？	116
8.6	如何知道串行发送/接收是否已经完成？	116
8.7	串行口如何控制？	117
8.8	复位后，单片机的串行口会自动处于接收数据的状态吗？	117
8.9	串行缓冲寄存器 SBUF 有什么作用？	118
8.10	什么是波特率？如何确定波特率？	118
8.11	如何计算串行方式 1、3 的波特率？	118
8.12	什么是串行发送？发送时数据传送的顺序是怎样的？	119
8.13	什么是串行接收？接收时数据传送的顺序是怎样的？	119
8.14	需要关心发送/接收的详细过程吗？	120
8.15	串行口的发送/接收会影响程序的执行吗？	120
8.16	串行方式 0 的发送/接收条件是怎样的？	120
8.17	串行方式 1 的发送/接收条件是怎样的？	121

8.18	串行方式 2、3 的发送/接收条件是怎样的？	121
8.19	串行方式 2、3 中 TB8、RB8 位起什么作用？	122
8.20	什么是主从通信方式？	122
8.21	串行控制寄存器 SCON 的 SM2 位有什么作用？	123
8.22	如何应用串行方式 1？	123
8.23	如何应用串行方式 2、3？	130
8.24	一个完善的单片机串行通信系统应考虑哪些因素？	137
第 9 章 单片机接口技术与存储器扩展		138
9.1	程序存储器不够用，怎么办？	138
9.2	数据存储器不够用，怎么办？	139
9.3	什么是总线？	139
9.4	什么是三总线结构？	139
9.5	如何处理空余的高位地址线？	140
9.6	什么是 I/O 接口和接口技术？	141
9.7	什么是接口芯片和端口？	141
9.8	接口电路有哪些组成部分？	141
9.9	并行 I/O 口不够用，怎么办？	142
9.10	串行 I/O 口不够用，怎么办？	143
9.11	用 RAM 器件扩展的存储器一定是数据存储器吗？	143
9.12	用 ROM 器件扩展的存储器一定是程序存储器吗？	143
9.13	如何扩展程序存储器，并对扩展电路进行分析？	144
9.14	如何扩展数据存储器，并对扩展电路进行分析？	145
9.15	如何用 EEPROM 扩展存储器，并对扩展电路进行分析？	147
9.16	如何访问扩展的程序和数据存储器？	149
9.17	如何理解 P0、P2 口作为 I/O 口和地址/数据总线时使用上的区别？	149
9.18	P0、P2 口能同时作为 I/O 口和地址/数据总线使用吗？	150
9.19	如何实现用外部数据存储器方式扩展 I/O 口？	150
9.20	单片机与外部器件间有什么数据传送方式？	152
第 10 章 显示与键盘接口技术		153
10.1	什么是 LED 数码管动态显示的“段控”和“位控”？	153
10.2	LED 数码管显示器有哪些工作方式？	154
10.3	要显示的数字、字符，如何变成显示的“段码”？	155
10.4	动态显示时如何保证同时清晰看到所有位？	155
10.5	如何更新显示的内容？	155
10.6	动态显示为什么采用显示缓冲区的形式？	155
10.7	如何编写动态显示程序？	156
10.8	以按键和键盘作为输入器件时，为什么需要消抖？	163
10.9	如何编写简单的按键处理程序？	163

10.10	矩阵键盘电路是怎怎样的？	166
10.11	单片机如何知道矩阵键盘何时有关键按下？	166
10.12	如何识别矩阵键盘中哪个键按下？	167
10.13	如何设计矩阵键盘处理程序？	168
10.14	一个完善的键盘处理程序应考虑哪些因素？	172
10.15	LCD 液晶显示模块有什么特点？	173
10.16	LCD 液晶显示模块有哪些引脚？	173
10.17	LCD 液晶显示模块有哪些寄存器？	174
10.18	LCD 液晶显示模块有哪些编程指令？	175
10.19	MCS-51 指令可以对 LCD 模块编程吗？	177
10.20	如何使用 LCD 液晶显示模块？	178
10.21	什么是 SPI 总线？	182
10.22	显示驱动器 MAX7219 有什么特点？	183
10.23	如何使用和设置 MAX7219 的寄存器？	185
10.24	如何应用 MAX7219？	186
10.25	带 SPI 接口的单片机如何连接和控制 MAX7219？	191
第 11 章	实时时钟与温度传感器接口技术	197
11.1	实时时钟 DS12C887 如何与单片机接口？	197
11.2	串行实时时钟 DS1306 有什么功能？	202
11.3	单片机如何与串行实时时钟 DS1306 接口？	204
11.4	什么是单总线？	208
11.5	温度传感器 DS18B20 有什么特点？	209
11.6	DS18B20 有哪些存储单元？	210
11.7	什么是循环冗余校验？	211
11.8	DS18B20 是怎样实现 CRC 校验的？	212
11.9	MCS-51 单片机如何连接和控制 DS18B20？	213
11.10	单片机控制 DS18B20 的应用程序举例	217
附录 A		223
	附一：MCS-51 单片机指令系统功能分类及注释表	223
	附二：MCS-51 单片机指令系统顺序查询及注释表	226
附录 B	部分程序索引	230
	附一：汇编语言程序	230
	附二：C51 语言程序	231
参考文献		233

第1章 对单片机的基本认识

众多单片机的教材，往往是以单片机的原理和一般介绍开篇的，基本不涉及对器件本身的感性认识。这使得初学者有一种不踏实的感觉，存在着“我所学的东西是个什么样子”的疑惑。

现在，单片机的应用日益广泛，开设单片机课程的学校和专业越来越多，单片机学习者的知识背景不尽相同，如有的之前学过微机原理，有的则没有学过，他们都有可能有这样的疑惑：“单片机的结构是怎么样”、“单片机和微机有什么区别”、“单片机与专用芯片有什么区别”。

本章简要地回答初学者在学习单片机时遇到的一些基本问题。

1.1 单片机是什么？

单片机是在一个集成电路芯片上，包含有中央处理器 CPU、程序存储器、数据存储器、并行和串行输入/输出端口、定时器/计数器、中断系统等功能部件的电子器件。单片机（Microcontroller），又称微控制器。

随型号的不同，有的单片机还包含有 A/D 转换器、液晶控制端口、不同类型的串行端口、D/A 转换器等功能部件。这些不同的功能，为我们使用单片机提供了丰富多彩的选择范围。

在微型计算机系统发展的早期，研发人员为了简化结构，把 CPU 和随机存取数据存储器（RAM）、只读程序存储器（ROM）、定时器/计数器、总线结构、简单的输入/输出接口等电路焊在同一块印制板上，构成了最简单的微型计算机，即单板机，用来控制各种工业设备和仪器仪表或进行教学实验。单片机名称的最初由来，就是相对单板机而言的。

1.2 单片机是什么样子？

市场上销售的单片机的外形一般有双列直插式和 PLCC 式封装，其尺寸与引脚数有关。图 1-1 是常见的两种单片机外形图。

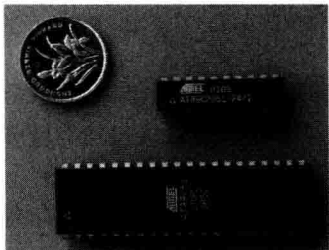


图 1-1 单片机外形

1.3 单片机有哪些型号?

单片机有不同系列,每一系列又有多种型号。教科书和相关材料都有很多这方面的介绍。读者可以自行查阅,也可以作为单片机课程的一项作业来完成。

目前,国内应用最多的是 MCS-51 系列及其兼容的单片机。MCS-51 系列是 Intel 公司研发的第二代单片机,其基本器件是 8051。由于 8051 的程序存储器是掩模型只读存储器 ROM,因此曾经广泛使用没有内部程序存储器的 8031 外接可擦写的只读存储器 EPROM 构成单片机应用系统。随着 Flash 闪速存储器在单片机上的应用,由 Atmel 公司出品、与 MCS-51 系列完全兼容、可以对程序存储器进行电擦写的 AT89C 系列单片机取代了 8051、8031 等产品,得到广泛应用。在把程序写入 AT89C 系列单片机的存储器时,仍需要使用专门的编程器。而现在,时兴的是 Atmel 公司的 AT89S 系列单片机。AT89S 单片机的系统内编程(In system programmable)功能,可以实现在应用系统上直接对单片机进行程序改写,而不必把单片机器件从系统中取出放在专门的编程器上改写程序,这是 AT89S 系列取代 AT89C 系列的重要原因。

1.4 从外形看,怎么知道单片机的功能?

单片机的外形,与一般的集成电路芯片无异。单纯地从单片机的外形上是看不出它的功能的。

1.5 怎么知道单片机是否在工作?

单片机是半导体有源器件,即必须接上电源才能工作。除了电源外,单片机还必须进行一些必要的连接后才能正常工作。单片机的工作,主要是输入外部信号,进行一定的处理后,控制其他的外接器件。从这些器件的工作情况,可以判断单片机的工作是否正常,如可以从单片机控制的指示灯是否按我们的期望进行指示来判断它是否在工作。

1.6 单片机和微机有什么区别?

单片机又称微控制器,其本身就是集成在一个芯片上、配置相对固定的计算机系统。单片机以占用空间小、耗能低、输入/输出操作能力强、接口功能较强、有较强的控制功能、可以灵活方便地构成应用计算机系统等为特点。

一般指称的微处理器,指的是通用目的微处理器,具有高速运算和强劲的数据分析与处理能力,如 Intel 公司的 x86 系列及奔腾系列及 Motorola 公司的 PowerPC 系列等。这些微处理器本身的芯片上并没有存储器和输入/输出端口,因此,它们通常被称为通用微处理器。

通用微处理器必须与外部加入的随机存储器 RAM、只读存储器 ROM、输入/输出端口、定时器等连接起来才能构成一个完整的计算机系统，即微机。一般微机系统具有体积较大、价格昂贵、功能多、计算能力强等特点。

1.7 单片机与专用集成电路有什么区别？

专用集成电路 ASIC (Application Specific Integrated Circuits)，是指按特定用户要求和特定电子系统的需要而设计、制造的集成电路。采用专用集成电路，可以使电路体积更小、重量更轻、功耗更低、可靠性和性能提高、保密性增强、成本降低。

可编程逻辑器件 PLD 是可以编程的方式实现和改变逻辑功能的专用集成电路器件，可以满足一般数字系统的要求。它所实现的逻辑功能由用户通过对器件编程来确定。目前，生产和使用的可编程逻辑器件品种主要有可编程阵列逻辑 PAL、通用阵列逻辑 GAL、可擦除的可编程逻辑器件 EPLD、复杂可编程逻辑器件 CPLD、现场可编程门阵列 FPGA 等。这些集成电路内部由基本的单元逻辑模块、输入/输出模块、实现互连的开关矩阵等部分组成，用专门的编程语言（如硬件描述语言 VHDL、Verilog HDL）来实现基本单元逻辑的连接，使硬件的功能可以像软件一样通过编程来修改，从而实现不同的功能。编程的数据存储在只读存储器 ROM 或随机存储器 RAM 中。不同类型的器件，在内部结构、集成度、速度及编程方式上具有各自的特点。

复杂可编程逻辑器件 CPLD 和现场可编程门阵列 FPGA 是用得比较普遍的可编程逻辑器件。它们是在 PAL、GAL 等逻辑器件的基础之上发展起来的，可以替代几十甚至几千块通用 IC 芯片，适合于开发周期短、有一定复杂性和电路规模的数字电路设计。这样的专用集成电路器件实际上就是一个子系统部件。

专用集成电路与单片机一样，都是由超大规模的集成电路构成的可编程器件，都使用了 ROM 或 RAM 形式的存储器。不同的是，单片机主要面向控制，而专用集成电路是从通用型数字集成电路发展而来的，主要是以硬件方式实现特定的逻辑功能。

1.8 单片机与可编程控制器（PLC）有什么区别？

可编程控制器（PLC）是在工业控制领域得到广泛应用的数字设备。国际电工委员会（IEC）于 1987 年对可编程控制器的定义如下：“可编程控制器是一种进行数字运算操作的电子系统，是专为在工业环境下应用而设计的工业控制器。它采用了可编程序的存储器，用来在其内部存储能执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字式或模拟式的输入和输出，控制各种类型机械的生产过程。可编程控制器及其有关外围设备，都按易于与工业系统联成一个整体、易于扩充其功能的原则设计。”

从以上定义可知，PLC 是专为工业环境下应用而设计的工业控制计算机。具有很强的抗干扰能力，以及广泛的适应能力，是 PLC 区别于其他微机系统的一个重要特性。

由于可编程序控制器实质上是一种工业控制计算机，其内部基本结构与一般微型计算机类似，主要由中央处理单元（CPU）、存储器、输入/输出接口和通信接口等部分组成。

PLC 常用的 CPU 有 3 种形式,即通用微处理器、单片机、位片式微处理器。

与个人计算机一样,PLC 有专门的电源模块。电源模块除为 PLC 提供工作电源外,还可同时向外部提供直流电源。

大多数的 PLC 编程系统均采用类似于继电器—接触器控制线路的梯形图语言编程,与常用的计算机语言相比,更容易被一般工程技术人员所理解和掌握。

PLC 是具有完整的机壳和接线端子的工业控制计算机成品,具有较强的 I/O 能力、扩展和通信能力,编程语言简单,但成本很高。而单片机从外形上看,是集成电路芯片,由于引脚数目和所集成的功能的限制,所具有的 I/O 功能或外部功能并不齐全,且驱动能力、抗干扰能力较弱,需要与其他电路一起才能构成功能较强的应用系统,但正是由于单片机只是一个器件,成本低廉,具有按照不同的需要构成不同应用系统的强大灵活性。

1.9 单片机的组成结构是怎样的?

各种计算机系统,不论是个人计算机还是面向控制的单片机,从硬件体系结构来说,都是由五大部分构成,即运算器、控制器、存储器、输入设备、输出设备,如图 1-2 所示,运算器在控制器的控制下,对信息及数据进行处理和计算。控制器将从存储器取出的程序指令进行译码和分析后发出各种控制信息,指挥各个部件的工作。存储器用来存放程序和数据。存储器在控制器的控制下,与输入设备、输出设备、运算器及控制器交换信息。输入设备用来输入程序和各种原始数据。输出设备用来输出计算和信息处理的结果。人们往往把运算器、控制器、存储器合称为计算机的主机,把输入设备、输出设备统称为外部设备(或外围设备)。在主机部分,又把运算器和控制器集中在一起称为中央处理单元——CPU。

单片机是高度集成化的电路元器件,在它的内部,除了有 CPU 以外,还集成了其他一些重要的功能部件。这些功能部件的作用与 CPU 的作用相互联系,共享芯片内的各种资源,所以不能清晰地区分哪一部分是 CPU,哪些是其他功能部件。因此,单片机的结构往往又按其功能划分,读者可以参阅单片机的教科书。

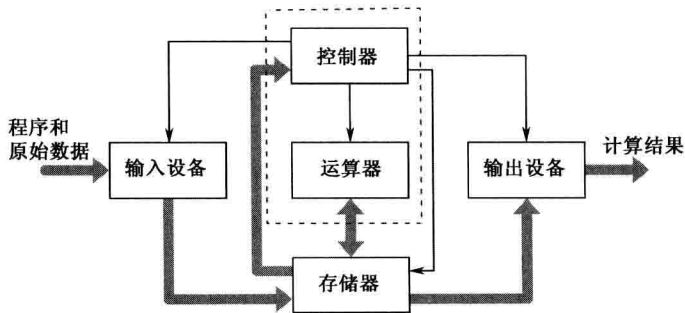


图 1-2 计算机的基本结构