



普通高等教育“十一五”规划教材

单片机原理与应用技术

高惠芳 主编



科学出版社
www.sciencep.com

普通高等教育“十一五”规划教材

单片机原理与应用技术

主 编： 高惠芳

副主编： 张海峰 胡 冀

崔佳冬 曾 毓

科 学 出 版 社

北 京

内 容 简 介

本书针对目前最通用的 MCS-51 单片机,在汇编语言的基础上,增加了目前最流行的 C51 程序设计语言,内容主要包括:单片机芯片的硬件原理和结构、汇编语言指令系统和程序设计、C51 的程序设计、单片机内部资源(包括中断、定时/计数器、串行口)、单片机系统扩展(包括存储器扩展、I/O 扩展)及功能扩展(包括键盘、显示器、A/D 及 D/A 转换)、Keil C 集成调试软件及 Proteus 仿真软件的使用介绍等。

本书的特点是通过汇编语言和 C 语言穿插进行讲述,实例较多,且很多例子都给出了汇编语言和 C 语言的对照程序,使读者能同时学习汇编语言和 C 语言,并使熟悉汇编语言的读者能更快地学好单片机 C51 程序设计。

本书可作为高等院校电类、机械类专业本科生的教材,也可作为函授教材或培训班教材。另外,本书可供从事单片机应用产品研发的工程技术人员及单片机爱好者参考。

图书在版编目(CIP)数据

单片机原理与应用技术/高惠芳主编. —北京:科学出版社,2010
(普通高等教育“十一五”规划教材)
ISBN 978-7-03-027114-3

I. 单… II. 高… 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 (2010) 第 055683 号

责任编辑:耿建业 裴… / 责任校对:张怡君
责任印制:赵 博 / 封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号
邮政编码:100717

<http://www.sciencep.com>

新 蕾 印 刷 厂 印刷

科学出版社发行 各地新华书店经销

*

2010 年 4 月第 一 版 开本:787×1092 1/16
2010 年 4 月第一次印刷 印张:21 1/2
印数:1—4 000 字数:494 000

定价:36.00 元

(如有印装质量问题,我社负责调换)

前 言

单片机出现至今已经有 30 多年的历史,单片机技术也历经了几个发展阶段。目前,单片机已渗透到生活中的各个领域,几乎很难找到哪个领域没有单片机的足迹。导弹的导航装置,飞机的各种控制仪表,计算机的网络通信与数据传输模块,工业自动化过程的实时控制和数据处理设备,广泛使用的各种智能 IC 卡,民用豪华轿车的安全保障系统,摄像机、全自动洗衣机的控制系统,以及程控玩具、电子宠物等,甚至全自动控制领域的机器人、智能仪表、医疗器械,都离不开单片机。因此,单片机的学习、开发与应用将造就一批计算机应用与智能化控制的工程师和科学家。科技越发达,智能化的东西就越多,对单片机学习的需求也日益增加。

目前,单片机方面的教材大都采用汇编语言的讲解和设计程序实例,但汇编语言学习困难。在实际应用系统开发调试中,特别是开发比较复杂的应用系统时,为了提高开发效率和使程序便于移植,很多时候采用 C 语言。C 语言不仅学习方便,而且同汇编语言一样能够对单片机资源进行访问,因此目前大多数院校在开设单片机课程时都引入 C 语言。但引入 C 语言后,就发现在选择教材时存在两方面的问题:有的教材注重于单片机的原理,只使用汇编语言;而另一些教材注重于 C 语言,一般面向开发,不讲原理,属于高级教程,不适合初学者。能兼顾汇编语言和 C 语言的教材非常少,而在实际使用中需要一本在学习单片机基本原理的同时能兼顾汇编语言和 C 语言两个方面的教材。在整个大学阶段,大多数学生学习的课程中,只有单片机这门课能接触到汇编语言,所以该门课旨在使学生在汇编语言概念的基础上学会单片机的编程。本书编写的目的是在讲述单片机基本原理的同时能兼顾汇编语言和 C 语言两个方面。所以本书在大多数的实例中,相同的功能用汇编语言和 C 语言分别编程实现,通过用汇编语言和 C 语言两个方面的编程对比,使学生能够有选择地掌握一种语言,并认识另一种语言。同时,为了提高学生应用设计的能力,本书还介绍了目前单片机接口常用的接口芯片,列举了几个简单的单片机应用系统开发实例。

本书共分 11 章。第 1 章主要介绍单片机的发展和应用领域;第 2 章介绍 MCS-51 单片机的结构和原理;第 3 章介绍 MCS-51 单片机的汇编语言指令系统;第 4 章介绍 MCS-51 单片机汇编语言程序设计;第 5 章主要介绍单片机 C51 程序设计基础;第 6 章介绍单片机内部资源,包括中断、定时/计数器,以及串行口;第 7 章介绍单片机系统扩展,包括存储器的扩展、I/O 的扩展,以及串行口的扩展;第 8 章介绍单片机功能扩展,包括键盘、显示器、A/D、D/A 等;第 9 章介绍单片机应用系统的开发与设计;第 10 章介绍 Keil C51 软件的使用;第 11 章介绍可视化仿真开发工具 Proteus。

本书的第 2、3、6 章由高惠芳编写,第 4、7 章由张海峰编写,第 1、9、11 章和第 5 章的后 3 节由胡冀编写,第 8 章由崔佳冬编写,第 5 章的前 4 节和第 10 章由曾毓编写。黄继业、杨翠容、张文超等老师给本书的编写提出了很多宝贵意见,研究生赵文静为本书插图

的绘制付出了很多辛劳。在这里向各位表示感谢！

虽然全体参编人员已经尽心尽力,但限于自身水平,书中不妥之处在所难免,希望各位专家和广大读者不吝指正。另外,书中有些资料来源于网上,使用时间已久,无法查证作者,如果本书引用了您的观点,请与编者联系,编者将尽快更正参考文献,并对此表示诚挚的感谢。

本书可提供电子课件,有需要者请与编者联系索取。

高惠芳 gaohuifang@126.com

编 者

2010年3月

目 录

前言

第 1 章 单片机概述	1
1.1 单片机的基本概念	1
1.1.1 什么是单片机	1
1.1.2 微控制器 MCU	2
1.1.3 如何使用单片机	3
1.2 单片机的发展概况	3
1.2.1 单片机的发展历史	3
1.2.2 MCS-51 单片机的发展	4
1.2.3 现阶段主流单片机系列简介	5
1.3 单片机的应用特点与应用领域	6
1.3.1 单片机应用特点	6
1.3.2 单片机应用领域	7
1.4 单片机的发展趋势	8
习题	10
第 2 章 MCS-51 单片机的结构和原理	11
2.1 MCS-51 单片机的基本组成	11
2.1.1 MCS-51 单片机的基本组成	11
2.1.2 MCS-51 单片机的封装与信号引脚	14
2.2 MCS-51 单片机的并行 I/O 端口结构	17
2.3 MCS-51 单片机的存储器结构	21
2.3.1 程序存储器	21
2.3.2 数据存储器	22
2.4 MCS-51 单片机的时钟电路与时序	28
2.4.1 时钟电路	28
2.4.2 时序	29
2.5 MCS-51 单片机的工作方式	32
2.5.1 复位方式	32
2.5.2 程序执行方式	35
2.5.3 单步执行方式	35
2.5.4 低功耗方式	35
2.5.5 EPROM 的编程和校验工作方式	37
习题	38

第3章 MCS-51 单片机的汇编语言指令系统	39
3.1 指令格式及其符号说明	39
3.1.1 指令格式	39
3.1.2 常用符号说明	40
3.1.3 指令的字节	40
3.2 寻址方式	42
3.2.1 立即寻址	42
3.2.2 直接寻址	43
3.2.3 寄存器寻址	43
3.2.4 寄存器间接寻址	43
3.2.5 变址寻址	44
3.2.6 相对寻址	45
3.2.7 位寻址	45
3.3 MCS-51 单片机指令系统	46
3.3.1 数据传送类指令	46
3.3.2 算术运算类指令	50
3.3.3 逻辑运算及移位类指令	55
3.3.4 控制转移类指令	58
3.3.5 位操作类指令	64
习题	66
第4章 MCS-51 单片机汇编语言程序设计	69
4.1 汇编语言的概述	69
4.1.1 汇编语言的特点	70
4.1.2 汇编语言的伪指令	70
4.2 汇编语言源程序的编辑和汇编	73
4.2.1 手工编程和汇编	73
4.2.2 机器编辑和交叉汇编	73
4.3 汇编语言程序设计	74
4.3.1 简单程序设计	75
4.3.2 分支程序设计	76
4.3.3 循环程序设计	80
4.3.4 数制转换程序	90
4.3.5 查表程序设计	93
习题	95
第5章 单片机 C51 程序设计基础	97
5.1 C51 数据与运算	97
5.1.1 C51 的数据类型	97
5.1.2 C51 数据的存储类型	98

5.1.3	8051 特殊功能寄存器的 C51 定义	100
5.1.4	8051 并行接口及位变量的 C51 定义	101
5.2	C51 运算符、表达式及其规则	102
5.2.1	(复合)赋值运算符	102
5.2.2	算术运算符	103
5.2.3	自增和自减运算符	103
5.2.4	关系运算符	104
5.2.5	逻辑运算符	105
5.2.6	位运算符	105
5.2.7	条件运算符	106
5.2.8	指针和地址运算符	106
5.2.9	优先级和结合性	107
5.3	C51 流程控制语句	108
5.3.1	C51 程序的基本结构及其流程图	108
5.3.2	选择语句	109
5.3.3	循环语句	112
5.4	C51 构造数据类型	115
5.4.1	数组	115
5.4.2	指针	116
5.4.3	结构体	120
5.4.4	共用体	122
5.4.5	枚举	124
5.5	函数	125
5.5.1	函数的定义	126
5.5.2	函数的调用	127
5.5.3	函数的嵌套调用与递归调用	129
5.5.4	中断服务函数	129
5.5.5	指向函数的指针变量	131
5.5.6	局部变量和全局变量	131
5.6	C51 的库函数	134
5.6.1	一般 I/O 函数 stdio.h	134
5.6.2	字符函数库 string.h	135
5.6.3	标准函数库 stdlib.h 及其他头文件	136
5.7	C51 模块化程序设计	136
5.7.1	基本概念	136
5.7.2	模块化程序开发过程	138
	习题	138
	第 6 章 单片机内部资源及编程	140

6.1 中断系统	140
6.1.1 中断概述	140
6.1.2 中断源	142
6.1.3 中断控制	143
6.1.4 中断响应过程	146
6.1.5 中断请求的撤除	148
6.1.6 中断程序设计	149
6.1.7 外部中断源的扩展	152
6.2 定时/计数器	154
6.2.1 定时/计数器的结构及工作原理	154
6.2.2 定时/计数器的控制	156
6.2.3 定时/计数器的工作方式	157
6.2.4 定时/计数器的初始化	160
6.2.5 定时/计数器应用举例	162
6.3 串行通信口	170
6.3.1 数据通信概述	171
6.3.2 单片机的串行通信接口	177
6.3.3 串行通信的工作方式及波特率设置	180
6.3.4 串行口应用举例	184
习题	194
第7章 单片机系统扩展	196
7.1 单片机最小应用系统	196
7.1.1 单片机最小应用系统构成	196
7.1.2 系统扩展的内容与方法	197
7.2 存储器的扩展	198
7.2.1 程序存储器的扩展	198
7.2.2 数据存储器的扩展	200
7.2.3 存储器综合扩展	200
7.2.4 闪速存储器及其扩展	201
7.3 输入与输出口的扩展	203
7.3.1 简单的并行 I/O 接口扩展	203
7.3.2 用 8255 扩展并行 I/O 接口	205
7.4 串行口扩展	211
7.4.1 I ² C 总线及其接口芯片	211
7.4.2 SPI 接口及其接口芯片	213
7.4.3 单总线接口及其接口芯片	215
7.4.4 Microwire 串行总线及其接口芯片	220
习题	222

第 8 章 单片机功能扩展	223
8.1 键盘输入及接口	223
8.1.1 键盘概要	223
8.1.2 键盘程序设计	224
8.2 显示器及其接口	227
8.2.1 LED 显示器	227
8.2.2 LCD 显示器	235
8.3 D/A 转换器的接口与应用	237
8.3.1 DAC 的转换原理及分类	237
8.3.2 并行接口 DAC	240
8.3.3 串行接口 DAC	245
8.4 A/D 转换器的接口与应用	248
8.4.1 ADC 的转换原理及分类	248
8.4.2 并行接口 ADC	249
8.4.3 串行接口 ADC	253
习题	256
第 9 章 单片机应用系统的开发与设计	257
9.1 单片机开发系统	257
9.1.1 单片机实验开发系统介绍	257
9.1.2 单片机实验开发系统的构成	257
9.1.3 单片机实验开发系统的使用	258
9.2 51 实验板	260
9.2.1 51 实验板介绍	260
9.2.2 51 实验板的 S51 ISP 下载线使用	264
9.3 单片机应用系统的设计	266
9.4 单片机应用系统举例	267
9.4.1 步进电机的驱动	267
9.4.2 超声波测距	272
9.4.3 LED 点阵屏显示系统	278
习题	286
第 10 章 Keil C51 软件使用介绍	287
10.1 Keil C51 软件安装	287
10.2 μ Vision3 集成开发环境	287
10.3 建立工程项目	288
10.4 程序举例	292
10.5 编译程序	292
10.6 程序调试	295
习题	302

第 11 章 可视化仿真开发工具 Proteus 介绍	303
11.1 Proteus ISIS 编辑环境介绍	303
11.2 进入 Proteus ISIS 编辑环境	306
11.2.1 电路原理图的设计流程	306
11.2.2 电路原理图的设计方法和步骤	306
11.3 单片机仿真	312
11.3.1 虚拟工具箱	312
11.3.2 Proteus 在单片机仿真中的应用	313
11.4 Proteus 与 Keil 整合构建单片机虚拟实验室	317
11.4.1 Keil 的 μ Vision2 集成开发环境	318
11.4.2 Proteus 与 Keil 整合的实现	322
附录 MCS-51 单片机指令汇总	326
附表 1 数据传送类指令	326
附表 2 算术运算类指令	327
附表 3 逻辑运算及移位类指令	328
附表 4 控制转移类指令	329
附表 5 位操作类指令	332
参考文献	333

第 1 章 单片机概述

单片机在现代电子设计领域应用相当广泛,存在于数不胜数的家用电器、工业控制设备、电子仪器等电子产品中。了解、掌握单片机的应用知识对于每一个电子工程师来说都是必不可少的。本章介绍关于单片机的基本概念,并简要介绍单片机的历史发展、现状和未来趋势。

1.1 单片机的基本概念

1.1.1 什么是单片机

单片机,顾名思义,就是指单片微型计算机,这个称谓也是比较通俗的。从这点上理解,所谓单片机就是在单个芯片上的一台微型计算机。当然,这种计算机比起普通台式机来说,功能上要弱很多,但基本的计算机结构还是很完善的。

那么,单片机的构成到底与普通计算机有何相似之处。首先来分析一下普通计算机的组成结构,如图 1.1 所示。

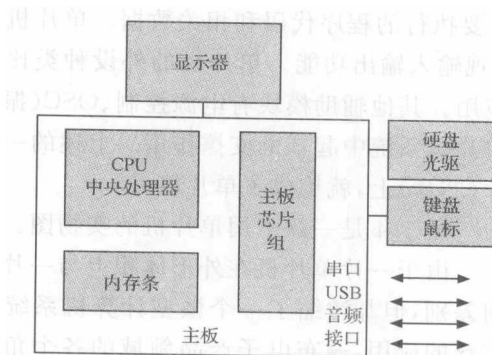


图 1.1 普通计算机的组成结构

普通计算机的核心部分往往由在主板上的中央处理器(central process unit,CPU)、主板芯片组、内存条,以及硬盘、光驱、显示器、键盘鼠标等构成。CPU 完成核心控制、计算功能,是解析执行程序代码的核心工作部件;主板芯片组是 CPU 与外界联系的通道和部分外设(外部设备)控制单元;内存条是存储设备,为 CPU 提供临时数据交换、临时程序代码存放功能,一般由 RAM(随机存储器)构成;光盘数据一般是只读的,可以理解为 ROM(只读存储器),硬盘用于存放永久或临时的程序代码和数据,光驱用于读取光盘数据,可以理解为 RAM、ROM 的复合体;显示器和键盘鼠标是人机接口,可以归为输入输出设备(I/O 设备)。经过上述分析,可以把图 1.1 的结构抽象成图 1.2 的结构。

不单是普通计算机,图 1.2 其实也是其他更复杂计算机的典型结构。本书将要详尽阐述的单片机也具有相同结构,如图 1.3 所示。单片机的主要部分也是由 CPU、RAM、ROM、外设、I/O 构成,只是相对于计算机系统各个单元都要简单很多,而且体积也小很多。

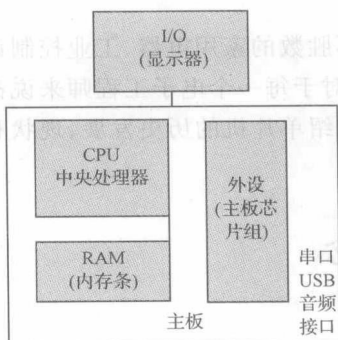


图 1.2 计算机组成结构的另外一种描述

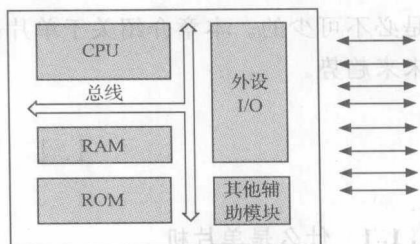


图 1.3 单片机的典型结构

单片机的 CPU 往往是专门设计的,为占用面积、成本、性能而优化,结构较为简单,运行速度较低,但同样可以执行程序代码,完成复杂控制。单片机的 RAM 一般容量都比较小,以降低占用硅片的面积,采用 SRAM(静态随机存储器)。ROM 的容量也较小,常常采用 flash(闪存)来实现,为降低成本也采用掩模(MASK)或者 OTP(单次可编程),ROM 里面一般存储 CPU 要执行的程序代码和相关数据。单片机的 I/O 较为简单,一般通过引脚的高低电平来实现输入输出功能。单片机的外设种类比较丰富,可针对不同的应用。其他辅助模块有电源控制、OSC(振荡器)、复位电路等,在单片机系统中起辅助支撑作用。上述的一切,都被放置在一个硅片(芯片)上,就构成了单片机。

图 1.4 是一款常用单片机的实物图。

由于一片单片机在外形体积上与一片普通的 IC 芯片没有任何差别,但却浓缩了一个微型计算机系统,所以一经出现就有了广泛的应用,遍布电子产品领域的各个角落。日常生活中,人们使用的各种电子装置大多有单片机的存在。

1.1.2 微控制器 MCU

由于大多数单片机的 CPU 较为简单,单片机往往不能胜任复杂的数学运算,而更着重于控制功能的实现。在绝大部分电子产品中,单片机是作为控制核心而出现的。因此“单片机”只是一个通俗的说法,更确切的称谓应该是微控制器(microcontroller)。

单片机的英文全称为 micro controller unit 或者 microcontroller,其英文缩写为 MCU。在本书中,“微控制器”、“MCU”、“单片机”都是同一个概念,无需区分。

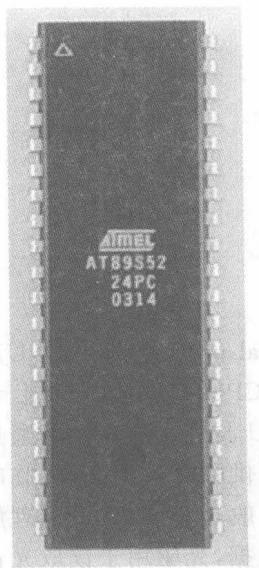


图 1.4 一款常用单片机的实物图

1.1.3 如何使用单片机

如前所述,单片机是一个单片化微型计算机系统,使用单片机完成一个具体应用需要对其编程,一般来说,flash结构的单片机在出厂的时候单片机内ROM(即flash)是空白的,可以认为它的功能未被定义,需要输入针对应用的代码来实现具体的应用功能。

为产生上述代码一般流程是需要在单片机开发环境当中编写针对程序,经编译获得所需执行代码。这里的单片机开发环境是指一种在计算机上的软件,通常它具有编辑程序、交叉编译、生成执行代码文件的功能。不同的单片机具有不同的开发环境,例如:图1.4所示的单片机属于MCS-51系列,可以使用Keil μ Vision 2这个集成开发环境,而AVR系列单片机却可以使用AVR Studio,除此之外,开发环境也可能支持不同的编程语言,常见的单片机开发语言有汇编语言和C语言。

单片机开发者需要针对具体的应用来写对应的汇编语言或C语言程序,在集成开发环境中进行编译、调试,最后获得所需的执行代码文件,通过编程器或其他特定的编程方式把执行代码文件烧写单片机的flash ROM中,这样该单片机就具有了针对应用的逻辑控制功能。

当然在开发过程当中,开发者的程序可能会存在问题,往往需要反复调试修改代码。这种调试一般有两种方式:一种是在集成开发环境中,用指令模拟器模拟单片机执行程序的过程;另一种是通过连接计算机与单片机系统的硬件调试工具(一般称为仿真器),在开发环境上进行硬件调试,正常情况下,调试时间远大于初始程序编写时间。

烧写了执行代码的单片机在系统上电后,就会按照开发者的意愿执行相应的逻辑控制功能。

1.2 单片机的发展概况

1.2.1 单片机的发展历史

在20世纪70年代初期,集成电路处于中小规模阶段,把一个微型计算机系统集成在单个芯片上,当时的集成电路的工艺不能满足。但开始出现单板机,即单板微型计算机,在一个电路板上实现一个微型的计算机系统。

1976年Intel公司推出了MCS-48单片机,这个可以认为是第一个真正意义上的单片机。从此,单片机技术迅猛发展,下面分阶段回顾单片机的发展历史,如图1.5所示。

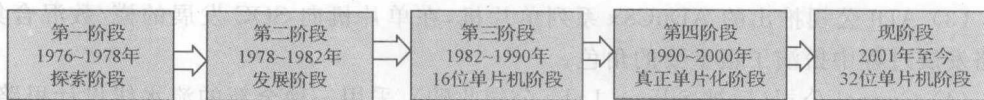


图 1.5 单片机发展历史

(1) 第一阶段(1976~1978年)。

Intel公司的MCS-48第一次采用了单片结构,即在一块芯片内含有8位CPU、定时/

计数器、并行 I/O 口、RAM 和 ROM 等。以体积小、功能全、价格低赢得了广泛的应用,为单片机的发展奠定了基础。

(2) 第二阶段(1978~1982 年)。

在 MCS-48 的带领下,各大半导体公司相继研制和发展了自己的单片机,如 Zilog 公司的 Z8 系列。到了 80 年代初,单片机已发展到了高性能阶段,如 Intel 公司的 MCS-51 系列、Motorola 公司的 6801 和 6802 系列、Rokwell 公司的 6501 及 6502 系列等。这一类单片机,如 MCS-51 系列,带有串行 I/O 口、8 位数据线、16 位地址线可以寻址的范围达到 64KB、控制总线、较丰富的指令系统等。

(3) 第三阶段(1982~1990 年)。

16 位单片机开始出现,除 CPU 为 16 位外,片内 RAM 和 ROM 容量进一步增大,实时处理能力更强,体现了微控制器的特征。此时的单片机均属于真正的单片化,大多集成了 CPU、RAM、ROM、多种的 I/O 接口、多种中断系统,甚至还有一些带 A/D 转换器的单片机,功能越来越强大,RAM 和 ROM 的容量也越来越大。可以说,单片机发展到了一个新的平台。这个时期单片机的种类较多,以 Intel 的 MCS-96 系列为代表。

(4) 第四阶段(1990~2000 年)。

新的高性能的单片机不断出现,新单片机普遍采用 RISC 架构,向高速、高集成度、数模混合、超低功耗方向发展。

(5) 现阶段(2001 年至今)。

32 位微控制器应用日益广泛,许多应用场合在单片机上大量使用 RTOS(实时操作系统),Embedded System 获得较大发展。在专用单片机发展方面,SOC 开始从概念走向实际应用。

1.2.2 MCS-51 单片机的发展

Intel 公司于 1980 年推出 8 位的 8051 单片机,称为 MCS-51 单片机,在单片机发展史上具有重要意义,当时在工业控制领域引起不小的轰动,并迅速确立了主导地位。后来,Intel 公司开放了 8051 单片机的技术,使得世界上很多半导体厂商加入到开发和改造 8051 单片机的行列中,其中产生较大市场影响的有:

(1) Atmel 公司,通过与 Intel 公司的技术交换,获得了 Intel 公司的 8051 制造技术,并且结合自己的 flash 技术,生产出了 ROM 为 flash 的 51 单片机,即 AT89 系列,使得单片机应用变得更灵活,在我国拥有大量的用户。

(2) Philips 半导体公司(现为 NXP),着力发展了单片机的控制功能和外围单元。

(3) ADI 公司推出的 ADuC8x 系列单片机,在单片机向 SOC 发展的模/数混合集成电路发展过程中扮演了很重要的角色。

(4) Cygnal 公司(后被 Silicon Labs 公司收购),采用一种全新的流水线设计思路,使单片机的运算速度得到了极大的提高,在向 SOC 发展的过程中迈出了一大步。

(5) Cypress 公司的 CY7C68 系列,是 MCS-51 单片机的改进版本,指令集有增强,主要用于 USB 接口应用。

目前 8051 单片机各生产厂商的主流产品有几十个系列,几百个品种。尽管其各具特

色,名称各异,但作为集 CPU、RAM、ROM、I/O 接口、定时器/计数器、中断系统为一体的单片机,其原理与结构大同小异。现以 Atmel 公司的 AT89S 系列产品为例,说明其系列之间的区别。

表 1.1 中列出了 Atmel 的 AT89S 系列(与 MCS-51 兼容)单片机在性能上略有差异。AT89S51 与 AT89S52 间所不同的是:片内程序存储器 flash 从 4KB 增至 8KB;片内数据存储器由 128B 增至 256B;定时/计数器增加了一个。

表 1.1 Atmel 的 AT89S 系列比较表

型号	flash /KB	RAM /B	F. max /MHz	I/O	UART	16 位 定时器	watchdog	封装
AT89S51	4	128	33	32	1	2	有	PDIP 40
								PLCC 44
								TQFP 44
AT89S52	8	256	33	32	1	3	有	PDIP 40
								PLCC 44
								TQFP 44
AT89S2051	2	256	24	15	1	2	—	PDIP 20
								SOIC 20

AT89S51 是兼容于 Intel 的 80C51,AT89S52 是兼容于 Intel 的 80C52。由此也可以看出 Intel 的 80C51 和 80C52 的区别。

AT89S2051 是 AT89S51 的 I/O 简化版本,可以使用的 I/O 为 15,但内部 SRAM 比 AT89S51 大倍,同时没有 watchdog 功能。

其他公司的 MCS-51 兼容单片机也与 Atmel 公司的一样,各个品种间有细微差别,让单片机的应用开发者根据实际情况选用。

1.2.3 现阶段主流单片机系列简介

前面介绍了 MCS-51 及其兼容系列的单片机,下面简要介绍一下现阶段国内主流单片机系列。为叙述方便,先以单片机 CPU 中 ALU(算数逻辑单元)的数据位宽为依据分成四类:4 位单片机;8 位单片机;16 位单片机;32 位单片机。

4 位单片机一般用于大量生产领域,主要是控制功能简单的电子玩具、家用电器等,开发时一般采用汇编(ASM)语言编写程序。4 位主流单片机这里不作介绍。

8 位主流单片机的种类很多,如下所示:

(1) Intel MCS-51 兼容单片机有很多,详细可以参考 1.2.2 节,是属于早期的 8 位单片机系列。

(2) Microchip PIC16C 5X/6X/7X/8X 系列、PIC17C、PIC18C 系列。

(3) Freescale 68HC908、68S08 系列。

(4) Atmel AVR 系列。

(5) 义隆 EM78 系列。

Microchip 的 8 位 PIC 系列拥有较大的市场份额,采用类 RISC 设计,在家用电器、工业控制上应用广泛。Freescale(前身为 Motorola 半导体)的 68 系列单片机具有高可靠性,广泛用于汽车电子领域。

16 位主流单片机,如下所示:

- (1) Intel MCS-96 系列,如 80C196。
- (2) TI MSP430 系列。
- (3) Microchip PIC24C 系列。
- (4) Maxim MAXQ 系列。
- (5) 凌阳 SPMC75 系列。
- (6) Freescale MC68S12 系列。

32 位主流单片机,如下所示:

- (1) ST STM32(Cortex-M3)。
- (2) Atmel AT32UC3B 系列(AVR32)。
- (3) NXP LPC2000 系列(ARM7 内核)。
- (4) Luminary Micro(TI 收购)的 Stellaris(群星)系列(ARM Cortex-M3 内核)。

1.3 单片机的应用特点与应用领域

1.3.1 单片机应用特点

关于单片机应用的特点可以分为以下几点。

1. 可应用于恶劣环境

在恶劣环境中,单片机构成的电子装置可以在程序的控制下,完成复杂的控制任务。单片机的高可靠性和强抗干扰能力,使它可以置于恶劣环境的前端工作。

2. 软硬件结合

单片机系统是一个软硬件结合的系统,系统的可伸缩能力较强。单片机系统是一个可以编程的系统,硬件上的不足有时可以通过软件的修改来满足控制要求。软硬件协同,可以处理较为复杂的现场控制问题。

3. 灵活定制

计算机系统的灵活性在单片机系统上也有体现。同一个单片机可以使用在完全不同的系统上。根据软件来决定系统功能,大大提高系统灵活性。

4. 升级方便

单片机通过使用 flash 技术使得程序的修改变得非常容易。在开发调试阶段,可以通过更新 flash 的内容,调试系统直到满足要求。当需要对产品进行升级时,也只要更新