

21 世纪计算机课程系列教材

# 微机系统原理及应用

主 编 毛玉良  
副主编 张赤斌  
参 编 田梦倩 王海燕  
许映秋 戴 敏

东南大学出版社

## 内 容 提 要

本书基础部分讨论了一般性问题,其中包括计算机中的数、码和常用电路,微处理器系统的基本结构和运行原理,微处理器、存储器和 I/O 的重要概念以及它们之间的相互关系。应用部分以 MCS-51 微控制器为主,介绍了它的结构特点,存储器组织,指令系统及集成的定时、中断和串行通信等功能。

围绕 MCS-51 的总线及其时序,介绍了存储器和外围芯片的扩展技术,并强调与时序配合。在存储器扩展部分介绍了各种译码方法。在 I/O 扩展方面,重点介绍了典型接口的工作原理及其应用,本书对传统的外围接口芯片及其功能进行了仔细的取舍。

在以 MCS-51 为基础的汇编语言教学中,引入 ASM-51 宏汇编的规范,它支持多模块以及今后同 C51 高级语言的混合编程。为了帮助初学者理解和掌握,汇编语言程序示例都尽可能完整地列出。本书例题丰富、重点突出、循序渐进,力求通俗易懂,体系和内容安排来自于作者多年从事的微机课程教学的经验和体会。书中大量例题来源于作者多年从事科研和工程技术的积累,实用性强,术语规范。

本书可作为普通高等学校非计算机专业本科学生的教材,也可作为成人高等教育的培训教材,还可供广大科技人员自学参考。

### 图书在版编目(CIP)数据

微机系统原理及应用/毛玉良主编. —南京:东南大学出版社,2006.8

(21 世纪计算机课程系列教材)

ISBN 7-5641-0520-8

I. 微... II. 毛... III. 微型计算机-高等学校-  
IV. TP36

中国版本图书馆 CIP 数据核字(2006)第 091794 号

### 微机系统原理及应用

---

出版发行 东南大学出版社

出版人 宋增民

社 址 南京市四牌楼 2 号

邮 编 210096

电 话 025-83792954 025-83793865(传真)

电 邮 Zhu\_min\_seu@163.com

---

经 销 全国各地新华书店

印 刷

开 本 787 mm×1092 mm 1/16

印 张 21

字 数 521 千字

版 次 2006 年 8 月第 1 版

印 次 2006 年 8 月第 1 次印刷

印 数 1—4000 册

定 价 35.00 元

---

# 前 言

本书是非计算机专业学生学习“微机原理及接口技术”的基础教材,主要是将 MCS-51 作为微控制器的一个典型系列,并围绕嵌入式系统的要求来阐述微机原理、技术及其应用。其出发点和编排与“MCS-51 单片机技术”类教材有明显的区别。最后一章也介绍了 80X86 的主要内容,可作为知识的拓展,避免读者产生以偏概全的认识错误。

本书面向初学者的需要,内容安排上增加了基础知识,同时考虑了实用性、实践性教学环节的可操作性以及有关专业学时上的限制。本书的主要特点是:

(1) 内容编排更加符合实际教学过程。目前许多教材内容的展开方式接近于其所介绍的微处理器(或微控制器)的权威资料,在具体教学过程中必须对其重组,才能通俗易懂,且给课后复习带来一定的困难。本书以初学者为对象,内容编排由浅入深,并随时结合实例应用,不断提高。这里也提醒读者在掌握基础知识后,应多看原始文档。

(2) 内容新颖。介绍了 Intel 的 ASM-51 宏汇编,因为它支持多模块编程。本书程序实例均按宏汇编格式编写,并尽可能完整。子程序大多按照 C51 高级语言的调用约定来写,为汇编与 C51 混合编程打下了基础。关于 C51 编程的介绍靠后,这样不会冲淡汇编语言的基础性,又使 C51 的实用技术介绍水到渠成。

(3) 强调学用一致。中断是本课程的难点,本教材突出中断技术的应用。从多年工程实践中我们体会到,几乎没有一个嵌入式程序开发可以不使用中断。中断编程要考虑的问题比较多,我们的办法是多给出一些中断服务程序的例子,这样初学者可作参照。

(4) 对常用外围芯片的内容进行了仔细的取舍。通过 TTL 电路构成输入缓冲,输出锁存,带联络信号的输入输出等,强化了接口原理;而 8255、8253 和 8250 则以原理、工作方式和典型应用为主,特别是已淘汰的 8250 以 16C2552 代替。至于这些芯片本身,随着 SOC 发展方向的到来,都是可取代的,或者被吸收到 MCU 中去。像 8255 的实际用量已经在大幅度下降,8250 已经停产,这是大势所趋。

本书由毛玉良副教授策划并任主编,张赤斌副教授任副主编。田梦倩、王海燕、许映秋、戴敏等老师参加了本书的编写。其中田梦倩老师编写了第 1 章;张赤

斌老师编写了第 2 章以及第 7、9、10、14 章的部分内容；许映秋老师编写了第 4 章以及第 12、13 章的部分内容；戴敏老师编写了第 15 章；王海燕老师编写了第 17 章；其余章节的内容由主编完成并统稿。沙洲工学院赵海兰老师，金陵科技学院郑李明、吴忠鸣老师也参加了部分章节的编写。研究生陈峰、吴洲、张耀宇、唐普亮等参与了资料收集、验证和部分文字工作，在此向他们表示感谢。

由于计算机技术日新月异的发展，新技术不断涌现，尽管我们在概念的阐述上已经考虑到这些因素，但仍然可能有不当之处，加之时间仓促，难免有错误发生，敬请各位读者、专家和使用本教材的老师批评指正，以便再版时及时改正。

编 者

2006 年 7 月

于南京

# 目 录

<b>0</b>	绪论 .....	(1)
<b>1</b>	计算机中的数和码 .....	(4)
1.1	有限字长的二进制数 .....	(4)
1.1.1	二进制数 .....	(4)
1.1.2	数制及其表示方法 .....	(5)
1.1.3	有限字长的二进制数 .....	(5)
1.1.4	常用术语解释 .....	(5)
1.2	十进制数到二进制数的转换 .....	(6)
1.3	带符号二进制数的表示及其运算 .....	(6)
1.3.1	带符号数的表示方法 .....	(6)
1.3.2	补码到真值的转换 .....	(8)
1.3.3	补码的运算 .....	(8)
1.4	溢出及运算的有效性 .....	(9)
1.4.1	无符号数与带符号数的溢出条件 .....	(9)
1.4.2	进位标志 CY 与溢出标志 OV .....	(10)
1.5	BCD 码 .....	(11)
1.6	ASCII 码 .....	(11)
<b>2</b>	常用数字电路 .....	(12)
2.1	逻辑电路 .....	(12)
2.1.1	与运算 .....	(12)
2.1.2	或运算 .....	(12)
2.1.3	非运算 .....	(12)
2.1.4	异或门 .....	(13)
2.1.5	累加器 .....	(13)
2.2	译码电路 .....	(13)
2.2.1	地址译码 .....	(13)
2.2.2	7 段数码管显示、译码 .....	(14)
2.2.3	简单的实验元件 .....	(14)
2.3	时序电路 .....	(15)
2.3.1	触发器 .....	(15)
2.3.2	锁存器 .....	(15)

2.3.3	分频器与计数器 .....	(16)
2.4	三态门和总线驱动 .....	(16)
2.4.1	三态门 .....	(16)
2.4.2	总线驱动 .....	(17)
2.5	数字电路设计中采用的先进技术 .....	(17)
2.5.1	PROM/PAL/GAL .....	(17)
2.5.2	CPLD/FPGA .....	(18)
2.5.3	数字系统中硬件方法与软件方法的特点与联系 .....	(19)
<b>3</b>	<b>微处理器</b> .....	(20)
3.1	微处理器系统的组成 .....	(20)
3.2	总线及总线时序 .....	(21)
3.3	关于微处理器的基本概念 .....	(23)
3.3.1	微控制器 .....	(23)
3.3.2	嵌入式微处理器 .....	(23)
3.4	CPU 的指令系统 .....	(24)
3.4.1	指令 .....	(24)
3.4.2	指令的编码 .....	(24)
3.4.3	指令系统 .....	(25)
3.4.4	指令的微操作 .....	(25)
3.4.5	指令的重叠执行 .....	(26)
3.4.6	执行速度方面的指标 .....	(26)
3.4.7	精简指令计算机 .....	(26)
3.5	CPU 中的寄存器 .....	(26)
3.5.1	寄存器与程序运行 .....	(27)
3.5.2	堆栈及堆栈指示器 .....	(28)
<b>4</b>	<b>存储器与存储管理</b> .....	(30)
4.1	存储器概述 .....	(30)
4.1.1	内存与外存 .....	(30)
4.1.2	半导体存储器的结构 .....	(30)
4.1.3	只读存储器 .....	(31)
4.1.4	随机访问存储器 .....	(32)
4.1.5	内存的主要性能指标 .....	(32)
4.1.6	新型存储器 .....	(33)
4.2	存储器的组织 .....	(33)
4.2.1	存储器映像 .....	(33)
4.2.2	存储器的地址空间 .....	(34)

4.2.3	哈佛结构 .....	(34)
4.3	存储管理 .....	(35)
4.3.1	逻辑地址与物理地址 .....	(35)
4.3.2	存储器的段式管理 .....	(35)
4.3.3	存储器的页式管理 .....	(35)
4.3.4	保护 .....	(36)
4.4	存储器扩展中的总线连接 .....	(36)
4.4.1	8位字长存储器的扩展 .....	(37)
4.4.2	16位字长存储器的扩展 .....	(37)
4.4.3	32位字长存储器的扩展 .....	(37)
4.4.4	Cache 高速缓冲技术 .....	(38)
<b>5</b>	<b>计算机的输入/输出 .....</b>	<b>(40)</b>
5.1	输入/输出的概念 .....	(40)
5.1.1	外部设备 .....	(40)
5.1.2	接口(Interface)与接口技术 .....	(40)
5.1.3	端口地址(Port Address) .....	(40)
5.1.4	端口及其编址方法 .....	(41)
5.2	输入/输出的数据传输方式 .....	(42)
5.2.1	程序传送方式 .....	(42)
5.2.2	中断概述 .....	(42)
5.2.3	中断传送方式 .....	(43)
5.2.4	DMA 方式的数据传送 .....	(44)
5.3	常用数字接口部件 .....	(45)
5.3.1	定时器/计数器 .....	(45)
5.3.2	并行口和串行口 .....	(46)
5.3.3	I <sup>2</sup> C 和 SPI .....	(46)
<b>6</b>	<b>MCS-51 系列微控制器 .....</b>	<b>(47)</b>
6.1	MCS-51 系列微控制器概述 .....	(47)
6.2	CPU 结构 .....	(48)
6.2.1	运算器 .....	(48)
6.2.2	布尔处理机 .....	(49)
6.2.3	控制器 .....	(49)
6.3	MCS-51 的存储器组织 .....	(51)
6.3.1	程序存储器 .....	(52)
6.3.2	内部数据存储器 .....	(53)
6.3.3	特殊功能寄存器 .....	(54)

6.3.4	外部数据存储器 and I/O .....	(56)
6.4	MCS-51 系列微控制器的引脚功能 P0P3 .....	(56)
6.4.1	P1 口的内部结构 .....	(57)
6.4.2	P3 口的内部结构 .....	(58)
6.4.3	P2 口 .....	(59)
6.4.4	P0 口 .....	(59)
6.5	单片方式以及总线扩展方式 .....	(59)
6.5.1	单片方式 .....	(59)
6.5.2	应用示例 .....	(60)
6.5.3	单片机的总线扩展 .....	(61)
<b>7</b>	<b>MCS-51 指令系统</b> .....	(62)
7.1	MCS-51 指令概述 .....	(62)
7.1.1	指令分类 .....	(62)
7.1.2	书写格式与缩略符 .....	(62)
7.1.3	寻址方式 .....	(63)
7.2	数据传送类指令 .....	(65)
7.2.1	内部数据传送指令 .....	(65)
7.2.2	栈操作指令 .....	(66)
7.2.3	外部数据传送指令 .....	(67)
7.3	运算类指令 .....	(68)
7.4	逻辑运算类指令 .....	(70)
7.5	控制类指令 .....	(72)
7.5.1	程序转移指令 .....	(73)
7.5.2	子程序调用和返回指令 .....	(77)
7.6	位操作指令 .....	(79)
7.6.1	位操作与位地址 .....	(79)
7.6.2	位传送指令 .....	(80)
7.6.3	位状态控制指令 .....	(80)
7.6.4	位逻辑运算指令 .....	(81)
7.6.5	基于位测试的控制转移指令 .....	(81)
<b>8</b>	<b>汇编语言程序设计</b> .....	(82)
8.1	汇编语言语句结构 .....	(82)
8.1.1	程序行的组成 .....	(82)
8.1.2	宏汇编中的常数 .....	(83)
8.1.3	宏汇编中的表达式 .....	(83)
8.2	伪指令 .....	(84)

8.2.1	段定义伪指令 .....	(84)
8.2.2	符号定义伪指令 .....	(85)
8.2.3	存储单元定义并初始化 .....	(86)
8.2.4	保留地址伪指令 .....	(87)
8.2.5	模块连接伪指令 .....	(89)
8.2.6	其他伪指令 .....	(89)
8.3	常用程序设计方法 .....	(90)
8.3.1	最大值、最小值问题 .....	(91)
8.3.2	多分支转移问题 .....	(92)
8.3.3	二进制数据转为 BCD 码的问题 .....	(94)
8.4	模块化程序设计 .....	(97)
8.4.1	ASM-51 编程环境下的文件类型 .....	(97)
8.4.2	各种文件的关系及生成流程 .....	(98)
8.4.3	由两个模块构成的程序示例 .....	(98)
<b>9</b>	<b>MCS-51 的定时器/计数器与中断系统 .....</b>	<b>(102)</b>
9.1	MCS-51 的定时器/计数器 .....	(102)
9.1.1	定时器/计数器的结构 .....	(102)
9.1.2	定时器/计数器的工作方式 .....	(103)
9.1.3	定时器/计数器 T0 和 T1 的应用实例 .....	(107)
9.2	增强型 MCS-51 的 T2 定时器 .....	(111)
9.2.1	控制寄存器 .....	(112)
9.2.2	T2 的 16 位自动装载方式 .....	(112)
9.2.3	T2 的 16 位捕获方式 .....	(113)
9.2.4	T2 的波特率发生器方式 .....	(114)
9.3	MCS-51 的中断系统 .....	(115)
9.3.1	中断源 .....	(115)
9.3.2	中断源的屏蔽 .....	(116)
9.3.3	中断的优先级 .....	(116)
9.3.4	中断向量 .....	(117)
9.3.5	中断响应的条件和过程 .....	(118)
9.3.6	中断服务程序的特点和编写要求 .....	(119)
9.3.7	数码管动态显示技术 .....	(120)
9.3.8	定时器应用的其他示例 .....	(126)
<b>10</b>	<b>MCS-51 的串行通信 .....</b>	<b>(128)</b>
10.1	串行通信基本概念 .....	(128)
10.1.1	通信方式 .....	(128)

10.1.2	串行通信的波特率 .....	(130)
10.1.3	异步串行通信的常用物理标准 .....	(130)
10.2	MCS-51 的串行通信控制器 .....	(131)
10.2.1	串行通信相关特殊功能寄存器用法介绍 .....	(132)
10.2.2	串行口的工作方式 .....	(133)
10.3	串行通信控制器的应用 .....	(137)
10.3.1	移位寄存器方式应用举例 .....	(137)
10.3.2	串行通信的应用举例 .....	(140)
10.3.3	接收中断的循环缓冲区 .....	(148)
10.3.4	多机通信的中断接收 .....	(149)
<b>11</b>	<b>MCS-51 的总线 .....</b>	<b>(152)</b>
11.1	MCS-51 系统的时钟和复位电路 .....	(152)
11.1.1	时钟电路 .....	(152)
11.1.2	复位电路 .....	(153)
11.2	MCS-51 的总线 .....	(155)
11.2.1	总线的构成 .....	(155)
11.2.2	总线的时序 .....	(156)
11.3	应用举例 .....	(162)
11.4	存储器访问中的等待 .....	(163)
<b>12</b>	<b>MCS-51 的存储器扩展电路 .....</b>	<b>(164)</b>
12.1	存储器扩展 .....	(164)
12.1.1	存储器结构 .....	(164)
12.1.2	存储器的地址容量扩展 .....	(165)
12.1.3	译码电路 .....	(166)
12.1.4	存储器在实际应用中的技巧 .....	(168)
12.1.5	存储器的时序要求 .....	(169)
12.2	程序存储器扩展 .....	(171)
12.2.1	扩展示例 .....	(171)
12.2.2	微控制器的 ISP、IAP 概要介绍 .....	(174)
12.3	扩充数据存储器 .....	(174)
12.4	DS12C887B 日历时钟的扩展 .....	(180)
12.5	集成的扩展数据存储器 .....	(186)
<b>13</b>	<b>I/O 接口电路及其扩展 .....</b>	<b>(187)</b>
13.1	简单并行口扩展 .....	(187)
13.1.1	简单的并行口输入 .....	(187)

13.1.2	带锁存的并行口输入 .....	(188)
13.1.3	简单的并行口输出 .....	(189)
13.2	8255 通用可编程并行接口芯片 .....	(190)
13.2.1	8255 简介 .....	(190)
13.2.2	8255 的工作说明 .....	(192)
13.2.3	8255 与 MCS-51 总线的接口和编程 .....	(194)
13.3	8253 可编程外围定时器/计数器 .....	(198)
13.3.1	8253 的内部结构和引脚 .....	(198)
13.3.2	8253 的控制字和工作方式 .....	(199)
13.3.3	8253 与 MCS-51 总线的接口和编程 .....	(202)
13.4	SC16C2552 通用 UART 通信控制器 .....	(205)
13.4.1	SC16C2552 的功能描述 .....	(205)
13.4.2	芯片的引脚功能和内部结构 .....	(205)
13.4.3	SC16C2552 的寄存器 .....	(207)
13.4.4	SC16C2552 的扩展电路 .....	(211)
13.4.5	SC16C2552 的编程举例 .....	(212)
13.5	液晶点阵屏的接口设计 .....	(216)
<b>14</b>	<b>模拟量的输入和输出 .....</b>	<b>(219)</b>
14.1	D/A 转换器 .....	(219)
14.1.1	D/A 转换器的基本原理 .....	(219)
14.1.2	D/A 转换器的主要技术指标 .....	(221)
14.2	D/A 的应用电路 .....	(221)
14.2.1	D/A 电路的单极性和双极性设计 .....	(221)
14.2.2	DAC0832 .....	(223)
14.2.3	12 位 DAC7541A 接口电路 .....	(225)
14.3	A/D 转换器 .....	(227)
14.3.1	A/D 转换器原理 .....	(227)
14.3.2	A/D 转换器的主要技术指标 .....	(229)
14.4	典型 A/D 转换器的接口设计 .....	(230)
14.4.1	8 位 8 通道 ADC0809 及其应用 .....	(230)
14.4.2	12 位 AD574B 及其应用 .....	(234)
<b>15</b>	<b>C51 高级语言编程 .....</b>	<b>(239)</b>
15.1	C51 的数据类型 .....	(239)
15.2	Keil C51 扩展关键字 .....	(240)
15.3	存储区域与存储模式 .....	(240)
15.3.1	存储区域 .....	(240)

15.3.2	存储模式 .....	(242)
15.3.3	指针变量 .....	(242)
15.3.4	绝对地址访问 .....	(243)
15.4	C51 的函数 .....	(244)
15.4.1	函数的声明 .....	(244)
15.4.2	函数参数与返回值 .....	(244)
15.4.3	函数的寄存器组 .....	(244)
15.4.4	中断函数 .....	(244)
15.5	C51 模块化程序设计 .....	(245)
15.6	C51 与 ASM-51 宏汇编语言的混合编程 .....	(250)
15.6.1	函数名的转换 .....	(250)
15.6.2	参数传递 .....	(250)
15.6.3	函数的返回值 .....	(252)
15.6.4	C 语言和汇编语言的接口示例 .....	(253)
<b>16</b>	<b>Keil 51 应用基础 .....</b>	<b>(255)</b>
16.1	$\mu$ Vision2 的界面 .....	(255)
16.2	编辑源程序文件 .....	(256)
16.2.1	编辑一个汇编语言程序 .....	(256)
16.2.2	编辑一个 C 语言程序 .....	(257)
16.3	创建工程文件 .....	(258)
16.3.1	新建工程 .....	(258)
16.3.2	打开已存在的工程 .....	(263)
16.4	汇编/编译和链接 .....	(263)
16.5	程序调试功能 .....	(264)
16.5.1	进入调试模式 .....	(265)
16.5.2	单步执行 .....	(265)
16.5.3	在线汇编 .....	(266)
16.5.4	断点设置 .....	(266)
16.5.5	观察窗口 .....	(267)
16.5.6	寄存器窗口 .....	(267)
16.5.7	串行窗口 .....	(267)
16.5.8	性能分析窗口 .....	(268)
16.6	实例分析 .....	(268)
16.6.1	实例一 .....	(268)
16.6.2	实例二 .....	(269)
16.7	在线仿真调试功能 .....	(274)

<b>17</b>	<b>8086/8088 微处理器体系结构</b> .....	(276)
17.1	8086/8088 微处理器 .....	(276)
17.2	8086 指令系统 .....	(283)
17.2.1	8086 的寻址方式 .....	(284)
17.2.2	8086 指令系统概要 .....	(286)
17.2.3	数据传送指令 .....	(287)
17.2.4	算术运算指令 .....	(288)
17.2.5	逻辑运算和移位循环指令 .....	(290)
17.2.6	串操作指令 .....	(291)
17.2.7	控制转移类指令 .....	(293)
17.2.8	处理器控制指令 .....	(298)
17.3	DOS 功能调用 .....	(299)
17.4	8086 的 I/O 与中断系统 .....	(301)
17.4.1	I/O 接口 .....	(301)
17.4.2	等待电路 .....	(303)
17.4.3	8086 的中断 .....	(303)
17.4.4	8086 的硬件中断 .....	(305)
17.4.5	8086 的中断管理 .....	(306)
	习 题 .....	(312)
	参考文献 .....	(320)

计算机技术的发展以及应用方面的普及速度,甚至使年轻的读者都难免感叹。在计算机知识的教学方面,观念也一直在发生着变化。

以往对计算机的分类,主要是取决于其功能和规模:比如大型、中型和微型计算机,或者根据用途分为通用与专用。但无论是用 CPU 的集成度、运算速度还是计算机的存储容量,都不能刻画不断变化着的计算机世界。将通用计算机分为主机、服务器和 PC 机,这在 Internet 时代,是比较贴切的。这种划分代表了需求和相应的处理能力。

与通用计算机不同的另一类计算机被称为嵌入式计算机。它不是以通用计算机的形态出现,而是将应用了计算机技术的芯片,嵌入到应用对象中去。如工业控制、武器装备、仪器仪表、个人通信终端、家用电器控制、娱乐设施或器件等,都大量使用嵌入式计算机技术。

1972 年,当第一个 4 位的微处理器在 Intel 公司诞生的时候,人们期望用它来制造手持计算器。当 8 位的微处理器诞生的时候,人们用它来制造台式的个人计算机。同时,其工业应用也开始了。也就是说,在早期,即 1980 年左右,微处理器并没有分成通用和嵌入式应用。

当 8086/8088 经 80286/80386 向更高的目标——Pentium 发展的过程中,人们发现微处理器在应用上的定位开始日益明显。Intel 公司 1983 年左右推出的 80186/80188 就是为嵌入式应用服务的。它将 8086/8088 CPU 与外围功能芯片集成在一起,定位于工业应用。一二十年前出产的数控机床里面就可能找到这样的芯片。日本公司运用 80188/80186 的技术生产芯片,推出了 V20 和 V40 等,大量应用到数控机床和其他领域。但这类芯片从没用于通用计算机的制造,所以并不广为人知。

从 80286 起,微处理器的设计者一开始就考虑它将来如何与操作系统结合,以生产出功能更强、更好用的通用计算机。从工作在通信行业的朋友那里能了解到,80386 甚至 Pentium 也可以直接应用在大容量的数字交换机上,但这种应用已经是非常专业的了,跟通用计算机主板的设计一样,必须依赖 Intel 公司的技术支持,否则就难以完成设计任务。

专业化、知识和技术在向高科技企业集中,这是大势所趋。80X86 体系的发展,集成了更多计算机领域的发展成就、更多的原理和技术。

除了 Intel 公司以外, Motorola、AMD 等公司在推动微处理器从 8 位发展到 16 位再到 32 位的过程中,都有精彩的故事。而现在微处理器在高端的发展方向是多核、64 位。不断提升的性能,一定能使我们的计算机变得更加优异,使我们得生活和学习更加方便。

让我们再来看一看微处理器在嵌入式领域的发展和應用。Intel 公司早在 1977/1978 年间,就利用其微处理器技术,开发了 8 位的 8048 微型控制器。经过改进,于 1981 年推出了 8051 系列微控制器。现在通用计算机的键盘中,就有一个基于 8048 的控制器。到 1983 年,又推出了 8096 系列的 16 位微控制器。

二十多年过去了,人们本来以为 8051 系列在流行了一阵以后也会如 8088 一样被淘汰。从教学角度,人们也不看好它。这个以实用为主的、集成了 CPU、存储器和典型功能器件的芯

片,使用十分灵活的器件,一边被我们广泛地用着,一边却不被我们看好。

随着时间的推移,嵌入式微处理器出现了 32 位,时钟频率也从几兆上升到几百兆。可是,以 MCS-51 为代表的 8 位微处理器的销量不减反升,8 位 MCU 市场一直以 16% 的年复合增长率在增长。到 2004 年,MCU 与 DSP 的销售额占整个半导体市场的 10% 以上,可谓半导体家族中的重要分支。MCU 在 2004 年的销售额达到 120 亿美元,出货量达 70 亿片。在 MCU 产品中,8 位 MCU 和 32 位 MCU 被认为是两大支柱产品,其中,8 位 MCU 虽然面临各种新生代产品的挑战,地位仍岿然不动,而 32 位 MCU 伴随新兴通信网络的应用,也开始进入快速增长期,成为众多企业竞争的焦点。

“在嵌入式系统发展的 20 年历程中,流行过多次 8 位 MCU 的‘被替代论’,例如 16 位机推出时,CPLD/FPGA 出现后,32 位 ARM 应用热潮中,都有人扬言它们会替代 8 位 MCU,然而他们都错了。”嵌入式领域专家、北京航空航天大学何立民教授介绍说。他分析了 8 位 MCU 长盛不衰的原因:虽然某些 32 位 MCU 的价位下降,使原先使用 8 位 MCU 勉强胜任的电子系统转而使用 32 位 MCU,但 8 位 MCU 也在新形势下不断寻找新出路,如向 SoC 内核或智能器件方向发展,从而拓展出新的发展空间。他还说道,“而且,在很多 32 位的系统中,也有许多 8 位 MCU 的智能控制单元。就拿通用 PC 机来说,它们的键盘控制就采用了 8 位 MCU。”所以说,8 位 MCU 的应用空间是无限的,它将永不衰败。

成本因素是 8 位 MCU 存在的重要理由之一。虽然目前市场上一些低端 32 位 MCU 的价格已经在 50 元以下,但 30 元以下的 MCU 产品中,仍然以 8 位 MCU 为主。10 元以下产品就只有选择 8 位 MCU。

除了成本,8 位 MCU 在市场上永葆活力的另一个重要原因源自于它的不断变革,这使得 8 位 MCU 具有很高的市场适应性。8 位 MCU 曾进行的变革被总结称为“三次革命”:第一次是 Philips 公司将 8 位 MCU CMOS 化,不仅降低了 8 位 MCU 的成本,还使它能采用各种封装形式,集成各种外设,从而不断丰富产品系列;第二次是 Atmel 公司将 Flash 技术引入 8 位 MCU,迅速降低了带存储器 MCU 产品的成本,大大简化了用户对 MCU 擦写的流程,使用户可以快速对产品程序进行修改,从而满足市场对更短开发周期的需求;第三次革命是 Philips、Atmel、Silicon Labs、ADI、TI 等公司使 8 位 MCU 成为一个片上系统,集成了数字、模拟、存储器、传感器、看门狗等多种技术,提高了产品的集成度。

Silicon Labs 公司改进了目前的 8051 核心,把它的运行速度提高到 100 MIPS,这一速度足以替代 16 位 MCU 甚至低端 DSP 产品。此外,他们的 MCU 可以集成 24 位 ADC 和 12 位 DAC、晶体振荡器和传感器等,方便了用户在某些领域的设计和制造,并可提高终端产品的可靠性。

这些变革都给 8 位 MCU 增添了新活力,预示它们还有更加广阔的发展空间。MCS-51 系列 8 位微处理器历经二十多年不被淘汰的事实,使我们相信,实用性强不见得就不典范;虽然功能器件过于灵活了,但用的人多了,也就有了规范,学习时还是有规律可循的。

MCS-51 系列的成功与 Intel 公司采取的技术开放性策略也是分不开的。目前,全世界有最著名的 70 余家半导体厂商获得了该系列内核的授权,生产了数以百计的、各具特色的该系列微控制器。包括其他优秀的微控制器在内,世界市场上目前有 4 千余种微控制器在销售。正是由于微控制器的普及应用,使得计算机技术深入到世界的每一个角落。

嵌入式处理器和微控制器与通用计算机中的 CPU,虽然功能定位有很大的差距,但它们

---

毕竟是同源的。对于初学者来说,从本书的 MCS-51 系列微控制器入手,可以了解计算机运行的基本原理、基本的应用电路、基本的嵌入式编程技术,并可应用于工程实践,也可为以后深入学习高端的计算机硬件知识打下坚实的基础。

在计算机内部,数字使用二进制编码,通常只在输入输出环节采用十进制形式。可执行的程序代码或其他非数字信息也都以二进制形式编码,这样便于存取。本章介绍二进制数的数据表达、常用编码方法以及运算要点。

## 1.1 有限字长的二进制数

### 1.1.1 二进制数

由数字电路构成的“0”、“1”状态是信息存储的基础。1位可表示信息量很小,但是随着位数的增加,数的表示范围迅速扩大。二进制采用位置表示法,展开式如下:

$$N_B = \sum_{i=-m}^{n-1} B_i \times 2^i$$

$B_i$  表示第  $i$  位,可取 0 或 1; $m$  表示小数位的位数; $n$  表示整数位的位数;

二进制的书写比较冗长,易出错,所以书写时常采用十六进制。它的展开式为:

$$N_H = \sum_{i=-m}^{n-1} H_i \times 16^i$$

$H_i$  取 09、A、B、C、D、E、F 中的数。 $m$  表示小数位的位数, $n$  表示整数位的位数。数字 015 用十进制、二进制和十六进制表示如表 1-1 所示。

表 1-1 数字 015 用各种数制表示

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

因为  $2^4 = 16$ ,所以并不需要另外设置十六进制的运算或控制部件。对照 1-1 表,每 4 位二进制数可记作 1 位十六进制数;反之,1 位十六进制数可以拆解为四位二进制数用以分析。因此,可以将十六进制数看作仅仅是二进制数的缩略记法。