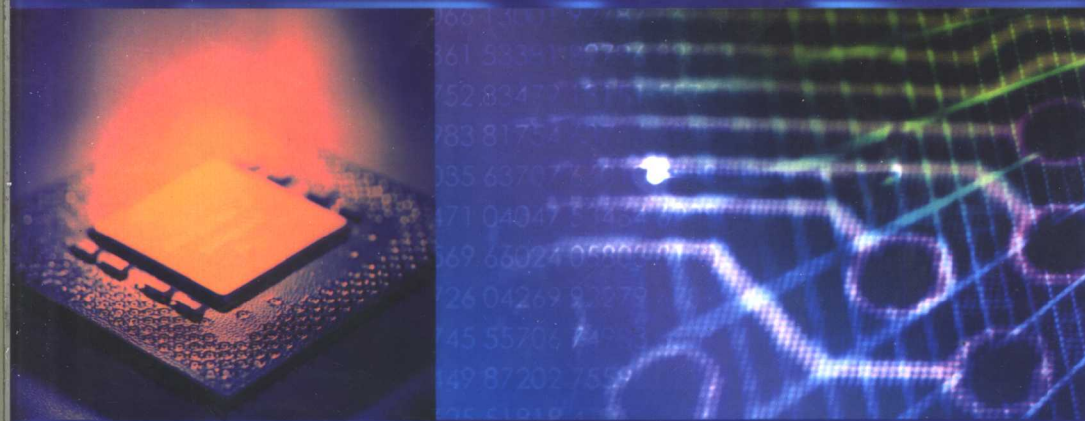




21 世纪高职高专规划教材

单片微计算机 原理、开发及应用



顾 滨 赵伟军 王 泰
鲍可进 李铁香



高等教育出版社

21 世纪高职高专规划教材

单片微计算机 原理、开发及应用

顾 滨 赵伟军 王 泰 鲍可进 李铁香

高等教育出版社

内 容 提 要

本书从最基本的基础知识入手,较详细地介绍了常用的 MCS-51 八位单片机应用技术,并对 AT 系列及 Intel 十六位机两种流行的单片机系统做了介绍。

全书共分八章。第一章介绍单片机的基础知识;第二章介绍单片机内部结构和原理、存储器的配置、并行输入/输出及时序;第三章介绍单片机指令系统及汇编语言程序设计;第四章讲述了单片机内有关输入/输出接口,包括中断、定时与计数、串行通信;第五章是 MCS-51 系统扩展技术。有存储器、串并行通信、人-机接口、A/D、D/A 及相应典型接口芯片的应用;第六章讲述软硬件开发过程,介绍了开发过程必备的设备、工具、环境及开发过程中应注意的问题;第七章讲述单片机在测控系统领域中的应用;第八章介绍了较新的具有 51 内核的单片机及 Intel 十六位单片机。

本书可作为高等职业、高等专科学校的教材,也可作为电子类维修培训班教材。

图书在版编目(CIP)数据

单片微计算机原理、开发及应用/顾滨等. -北京:
高等教育出版社 2000.8.(2003 重印)

ISBN 7-04-007306-7

I. 单… II. 顾… III. 单片微型计算机-基本知识
IV. TP368.1

中国版本图书馆 CIP 数据核字(2000)第 64948 号

单片微计算机原理、开发及应用

顾 滨 赵伟军 王 泰 鲍可进 李铁香

出版发行 高等教育出版社
社 址 北京市东城区沙滩后街 55 号
邮政编码 100009
传 真 010-64014048

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
排 版 高等教育出版社照排中心
印 刷 北京印刷三厂

开 本 787×1092 1/16
印 张 21.5
字 数 510 000

版 次 2000 年 8 月第 1 版
印 次 2003 年 5 月第 4 次印刷
定 价 21.50 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

本书是根据教育部全国高等工程专科计算机课程委员会九五选题计划和单片机的教学基本要求编写，由全国高等工程专科计算机课程委员会组织评审、推荐出版的规划教材。

本书在编写中力求做到：基本理论以必须、够用为原则，侧重于单片机实用性技术及实际应用的介绍，以操作、使用为主。在内容安排上，一方面紧密结合当前单片机新技术迅速发展的形势，介绍先进的 Flash 单片机及嵌入式技术，抛弃或减少将逐步淘汰的陈旧的技术内容；另一方面介绍应用上较普及的系统设计和应用技能。叙述以点带面、深入浅出、通俗易懂，使学生真正学有所用。

本书从最基本的基础知识入手，较详细地介绍了常用的 MCS-51 8 位单片机应用技术，并对 ATMEL 89 系列及 Intel 16 位机两类流行的单片机系统做了叙述。

全书共分八章。第一章讲述单片机的基础知识及其发展；第二章介绍单片机内部结构和原理、存储器的配置、并行输入/输出及时序。本章以在我国使用很普遍的 Intel 51 系列单片机为例，重点介绍其控制、存储部件及并行 I/O 口。这些都是单片机的重要内容，应该熟练掌握；第三章介绍单片机指令系统及汇编语言程序设计。它是基于第二章硬件知识之后的软件知识，也是重要的章节；第四章讲述了单片机内有关输入/输出接口系统，包括中断、定时与计数、串行通信。这些都是微机不可缺少的组成部分，同时又突出体现了“单片”的特点，因此其重要性是显而易见的；第五章是在第四章内容之上的片外扩展，有存储器、串行通信、人-机接口、A/D、D/A 及对应的典型接口芯片。这是开发单片机应用系统的必备知识。至此，应该将二、三、四、五章内容融汇贯通，学到单片机乃至基本微机系统的精髓；在第六章中规范了软硬件开发过程，介绍了开发过程必备的设备、工具、环境及开发过程中应注意的问题，这是很多经验型开发的理论依据。通过本章的学习，可使单片机应用人员掌握较全面的实践技能；第七章以单片机在测控系统领域的应用为例，从理论到实践介绍单片机的开发应用，是全书知识的总结和实现；第八章则介绍了全新的也是开发人员乐于采用的两种单片机——Intel 196 系列及 Flash 系列以开拓视野，为采用更新更先进的单片机来开发应用系统打基础。

本书由顾滨主编，赵伟军任副主编，第一、四章由王泰、顾滨编写，第五章由王泰编写，第二章由李铁香、赵伟军编写，第六章由顾滨编写，第七章由鲍可进、顾滨编写，第三、八章由赵伟军编写。顾滨负责了全书的统稿。

本书在编写、出版过程中得到了南京工程学院康乃真教授及高等工程专科计算机课程委员会的指导和帮助，在此表示衷心感谢。计算机技术发展迅速，加之编者水平有限、时间仓促，书中难免有疏漏之处，敬请批评指正。

编者

2000.3

目 录

第一章 导论	1	3.1 汇编语言与指令系统简介	36
1.1 概述	1	3.1.1 汇编语言程序设计的重要性	36
1.1.1 单片机及其发展	1	3.1.2 MCS-51 系列单片机的指令系统	36
1.1.2 单片机系统的组成	3	3.1.3 寻址方式	39
1.2 计算机中数的表示及运算	4	3.2 数据传送类指令	42
1.2.1 常用数制及转换	4	3.2.1 通用传送指令: MOV	44
1.2.2 机器数及其编码	8	3.2.3 外部数据存储器(或外扩 I/O 口) 与 A 传送指令: MOVX	48
1.2.3 字符信息的表示	12	3.2.3 程序存储器向累加器 A 传送指令: MOVC	49
习题	12	3.2.4 数据交换指令	50
第二章 MCS-51 单片机结构原理	14	3.2.5 栈操作指令	51
2.1 概述	14	3.3 算术运算类指令	52
2.2 MCS-51 单片机内部结构及引脚	15	3.3.1 加减运算指令	53
2.2.1 MCS-51 单片机内部结构	15	3.3.2 乘除运算指令	56
2.2.2 MCS-51 单片机外部引脚	16	3.3.3 增量、减量指令	58
2.3 MCS-51 单片机存储器配置	20	3.3.4 二-十进制调整指令	59
2.3.1 MCS-51 的存储器结构	20	3.4 逻辑操作类指令	60
2.3.2 程序存储器	21	3.4.1 对累加器 A 进行的逻辑 操作指令	61
2.3.3 内部数据存储器	21	3.4.2 双操作数指令	63
2.3.4 专用寄存器——特殊功能 寄存器 SFR	22	3.5 控制转移类指令	64
2.3.5 位地址空间	25	3.5.1 程序转移指令	64
2.3.6 外部数据存储器	26	3.5.2 子程序调用和返回指令	72
2.4 时序	26	3.6 位操作类指令	74
2.4.1 时钟的基本概念	26	3.6.1 位数据传送指令	75
2.4.2 典型指令的取指/执行时序	27	3.6.2 位状态控制指令	76
2.4.3 振荡源及控制器	28	3.6.3 位逻辑操作指令	77
2.5 并行 I/O 端口	29	3.6.4 位条件转移指令	78
2.5.1 P0 口	29	3.7 汇编语言及汇编过程	81
2.5.2 P1 口	30	3.7.1 汇编程序功能	81
2.5.3 P2 口	31	3.7.2 伪指令	82
2.5.4 P3 口	32	3.7.3 汇编程序的汇编过程	84
2.6 8031 最小系统	33	3.8 汇编语言程序的基本结构及 综合举例	86
习题	35	3.8.1 汇编程序的基本结构	86
第三章 MCS-51 指令系统及 汇编语言程序设计	36	3.8.2 综合举例	87

习题	96	5.4.2 按键、键盘及其接口	179
第四章 MCS-51 的片内接口	97	5.4.3 LED 显示器及其接口	190
4.1 MCS-51 中断系统	97	5.4.4 单片机应用系统中的典型键盘、 显示接口	199
4.1.1 中断的有关概念	97	5.5 模拟量与数字量转换接口基础	201
4.1.2 中断请求源	98	5.5.1 D/A 转换器接口	202
4.1.3 中断控制	100	5.5.2 A/D 转换器接口	211
4.1.4 MCS-51 中断的响应过程	101	习题	220
4.1.5 MCS-51 的外部中断	102	第六章 单片机开发系统及软件设计	221
4.2 定时/计数器	105	6.1 单片机开发系统	221
4.2.1 定时/计数器的结构	105	6.1.1 单片机的开发及开发工具	221
4.2.2 定时/计数器的 4 种工作模式	106	6.1.2 在线仿真技术与仿真器介绍	225
4.2.3 MCS-51 定时/计数器对 输入信号的要求	109	6.1.3 开发系统应用	226
4.2.4 定时/计数器的应用	110	6.2 单片机软件设计方法及工具	233
4.3 MCS-51 单片机串行接口及其应用	115	6.2.1 软件设计方法	233
4.3.1 串行通信概述	115	6.2.2 软件开发工具与环境	243
4.3.2 MCS-51 单片机串行接口	118	6.2.3 MCS-51 C 语言应用程序 开发技术	249
4.3.3 MCS-51 串行口应用	125	习题	254
习题	141	第七章 单片机在测控系统领域的应用	255
第五章 MCS-51 系统扩展技术	142	7.1 概述	255
5.1 MCS-51 单片机最小系统	142	7.1.1 工业控制机的组成	255
5.1.1 8051/8751 硬件最小系统	142	7.1.2 计算机控制系统的 基本结构模式	256
5.1.2 8031 硬件最小系统	142	7.1.3 工业控制机系统的分类	257
5.2 存储器的扩展	143	7.2 测控系统信息测量与处理	261
5.2.1 扩展程序存储器	143	7.2.1 单片机组成的测控系统	261
5.2.2 扩展数据存储器	148	7.2.2 开关量输入信号调理	262
5.2.3 外部扩展较大容量的 RAM	151	7.2.3 开关量输出驱动电路	263
5.2.4 新型存储器芯片在 MCS-51 系统中的应用	153	7.2.4 模拟量数据采集	264
5.3 I/O 接口的扩展	160	7.3 测控系统分析与设计	273
5.3.1 I/O 扩展概述	161	7.3.1 测控系统的准备阶段	273
5.3.2 扩展可编程并行 I/O 接口 8255A	162	7.3.2 测控系统的设计阶段	275
5.3.3 扩展可编程 RAM/I/O 接口芯片 8155	168	7.3.3 测控系统的仿真及调试阶段	279
5.3.4 用 TTL 芯片扩展 单片机的 I/O 口	174	7.4 测控系统的可靠性设计	280
5.3.5 用串行口移位寄存器方式扩展 并行 I/O 接口	176	7.4 由单片机构成的测控系统 示例及分析	283
5.4 常用通道配置与接口技术	177	7.4.1 水塔水位控制	283
5.4.1 单片机应用系统中的 人-机通道	178	7.4.2 交通信号灯模拟控制	286
		7.4.3 单片机数字程序控制系统实例	291
		习题	314

第八章 具有 51 内核的单片机及	
Intel 16 位单片机	299
8.1 89、90 系列 Flash 单片机	299
8.1.1 ATMEL 89 系列单片机	299
8.1.2 90 系列单片机	302
8.1.3 89 系列单片机系统 结构及存储器	304
8.1.4 指令系统	309
8.1.5 AT89C51 单片机 编程器的应用	310
8.2 Intel 16 位单片机	313
8.2.1 Intel 16 位单片机 系列芯片综述	313
8.2.2 8096 简介	313
8.2.3 80C196KB 简介	316
8.2.4 80C196KC 简介	320
8.2.5 80C196MC/MD 简介	326
附录	328
附录 1 ASCII 码表 (7 位码)	328
附录 2 MCS-51 系列单片机指令表	329

第一章 导 论

1.1 概 述

1.1.1 单片机及其发展

计算机的发明是 20 世纪最重大的科学技术成就之一，它使人类文明进入了一个崭新的时代。由于计算机具有强大的运算能力、逻辑判断能力、数据处理能力，因此自问世以来得到了飞速的发展。1971 年 Intel 公司制造出第一片微型机芯片 4004 之后，开始了计算机发展的另一个时代——微型机时代。微型机的发展速度非常迅猛，其核心部件——微处理器的性能和集成度几乎每两年增加一倍，而且愈来愈快。人们凭借先进的制造技术，可在一片面积很小的芯片上，构成中央处理器 CPU，这就是微处理器芯片。微处理器芯片、存储器（ROM、RAM）、输入/输出（I/O 口）及其他功能部件如定时/计数器、中断系统等通过地址总线（AB）、数据总线（DB）、控制总线（CB）结合在一起，构成了微型计算机系统的硬件部分。微型计算机的系统结构如图 1.1 所示。

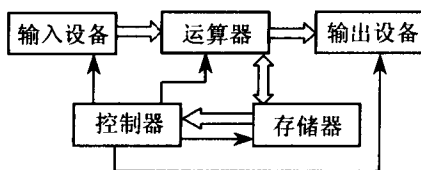


图 1.1 电子计算机的组成

这种计算机系统往往由多块印刷电路板构成，我们称它为多板机。如把它们集中做在一块印刷电路板上，便是实验室中常见的单板机了。多板或单板微型计算机体积大、功耗大、可靠性差，不宜在仪器仪表或其他需要微型化的场合应用。随着诸多应用领域的迫切需要，美国的仙童公司制造出了 F8 单片微型计算机，它把构成微型机的主要部分集成在一片半导体芯片上，这就是单片微型计算机（Single-Chip Microcomputer），简称单片机。

单片机是应工业测控的需要而诞生的，它的结构与指令功能都是按照工业控制要求设计的，故又叫单片微控制器（Single Chip Microcontroller），它使计算机系统以单片的形态出现。以单片机为核心的应用系统通常以最小系统运行，在家用电器、智能仪器仪表中常常可以看到“单片机”在工作。在工业测控系统中，往往在单片机的基础上外扩存储器、I/O 口以及一些外围电路，形成功能更强、更完善的系统。正是单片机体积小、功能强、可靠性高和价格低等优点，使得它被广泛应用于家电、工业过程控制、仪器仪表、智能武器、航空、汽车等领域。目前，世界上单片机的年产量已达十亿片，通常是微处理器产量的 4 倍以上。其数量之大和应用面之广是其他任何类型的计算机所无法比拟的。

单片机具有以下特点：

(1) 受集成度限制, 片内存储器容量较小。一般 ROM 在 8 KB 以下, RAM 在 256 KB 以内, 但通常可以用外部扩展的方法来扩大 ROM 或 RAM 的容量。

(2) 可靠性高。芯片本身是按工业测、控环境要求设计的, 其抗工业噪声干扰优于一般通用 CPU。程序指令及常数、表格固化在 ROM 中不易被破坏。许多信号通道均在一个芯片内。故可靠性高。

(3) 易扩展。片内具有计算机正常运行所必需的部件。芯片外部有许多供扩展用三总线及并行、串行输入、输出管脚, 很容易构成各种规模的计算机应用系统。

(4) 控制功能强。为了满足工业控制要求, 一般单片机的指令系统中均有极丰富的条件分支转移指令、I/O 口的逻辑操作以及位处理功能。一般说来, 单片机的逻辑控制功能及运行速度均高于同一档次的微处理器。

(5) 一般单片机内无系统通用管理软件, 只放置用户调试好的应用程序。

从最初的单片机到如今的新一代单片机, 其发展历史大致可分为四个阶段:

第一阶段 (1974 年~1976 年) 单片机的初级阶段。因受工艺技术水平限制, 单片机结构和功能都很简单。例如仙童公司生产的 F8 单片机, 内部仅有 8 位 CPU, 64 字节 RAM 和 2 个并行口, 还需一些其他芯片才能组成一台完整的微型机。

第二阶段 (1976 年~1978 年) 低性能单片机阶段。以美国 Intel 公司的 MCS-48 为代表。该系列单片机内集成有 8 位 CPU、并行 I/O 接口、一个 8 位的定时/计数器、片内 64 或 128 字节 RAM, 程序存储器空间最大 4 KB, 但无串行通信口, 不宜多机应用。

第三阶段 (1978 年~1983 年) 高性能单片机阶段。这一阶段推出的单片机虽然仍采用 8 位 CPU, 但均有多级中断功能、串行通信接口、16 位的定时/计数器, 而且片内 ROM、RAM 容量加大, 寻址空间范围可达 64 KB, 有的片内还带有 A/D 转换器接口。这一阶段单片机的生产厂家众多, 产品系列也特别多, 主要的有 Intel 公司的 MCS-51, Motorola 公司的 6801 和 Zilog 公司的 Z8 等。这类产品由于具有优异的性能价格比, 因而获得了极其广泛的应用。尤其是 MCS-51 系列单片机, 特别适合于控制应用, 在我国教育和经济建设等各个领域大显身手, 经久不衰, 是我国单片机应用的主流系列。

第四阶段 (1983 年~) 暂且称作新一代单片机阶段。这一阶段单片机的最重要标志是“单片机”的含义已发生了根本改变。目前大家仍然保留单片机这一习惯叫法, 但大家都明白它是指“Single-Chip Microcontroller”。新一代单片机有如下几方面的特点:

1) 内部 CPU 仍然以 8 位机为主流, 并不断完善。另一方面发展了 16 位、32 位的单片机。早期单片机多使用 CISC 体系结构, 近来 RISC 单片机已大力发展。在 RISC 单片机中, 除了指令集精简外, 一般都用单字长指令和流水线操作方法, 大大加快了单片机的指令运行速度。

2) 片内程序存储器有 ROM 型、EPROM 型、无 ROM (Romless) 型和低成本的 OTP (一次性烧入 ROM) 型。近年来, Flash ROM 已获得了普遍发展。由于 Flash ROM 可在线多次写入, 有些公司称之为 MTP (M-Time Programmable ROM)。Flash ROM 的普遍使用, 也导致了 ISP (在线可编程 In-System Programmable) 技术的迅速发展。可扩展容量有的已突破 64 KB, 达到 2 MB 以上。片内 RAM 已有 2 KB 以上的产品, 可扩展容量也可达到 2 MB 以上。

3) 芯片日趋复杂化、多样化、产品专用化。单片机在原来微型机的结构基础上, 集成嵌入了一些外设与一些外设驱动电路单元, 如通用接口、看门狗 (Watchdog)、A/D 和 D/A、LCD 驱动单元、遥控键盘、语音接口、串行总线²C, 甚至通用串行总线 USB (Universal Serial

Bus)、控制器局域网 CAN (Controller Area Network) 等。某些公司把数字信号处理器 DSP (Digital Signal Processor) 也嵌入了“单片机”内。

4) 多采用 CMOS 工艺, 出现双时钟、低电压单片机, 大大降低了系统功耗。时钟速度大幅度提高, 有的已达 100 MHz 以上。芯片引脚数过去多为 40 PIN, 现在已向两个方向发展——多引脚 (100 PIN 以上) 和少引脚 (20 PIN 以下)。封装形式多样化, 有 DIP、SH-DIP、QFP、SQFP、HQFP、TQFP、PGA、BGA、PLCC 等。

纵观单片机的发展历程, 单片机已经由纯粹的单片微型机到增加一定接口以适用于一般控制应用而成为单片微控制器。为了适应不同用户的各种各样的专门应用, 在通常的 CPU 内核之外又增加了相适应的特有功能的外设驱动器, 单片机又发展成了嵌入式微控制器。

1.1.2 单片机系统的组成

单片机系统的组成可用图 1.2 来概括, 它包括硬件和软件两大部分。

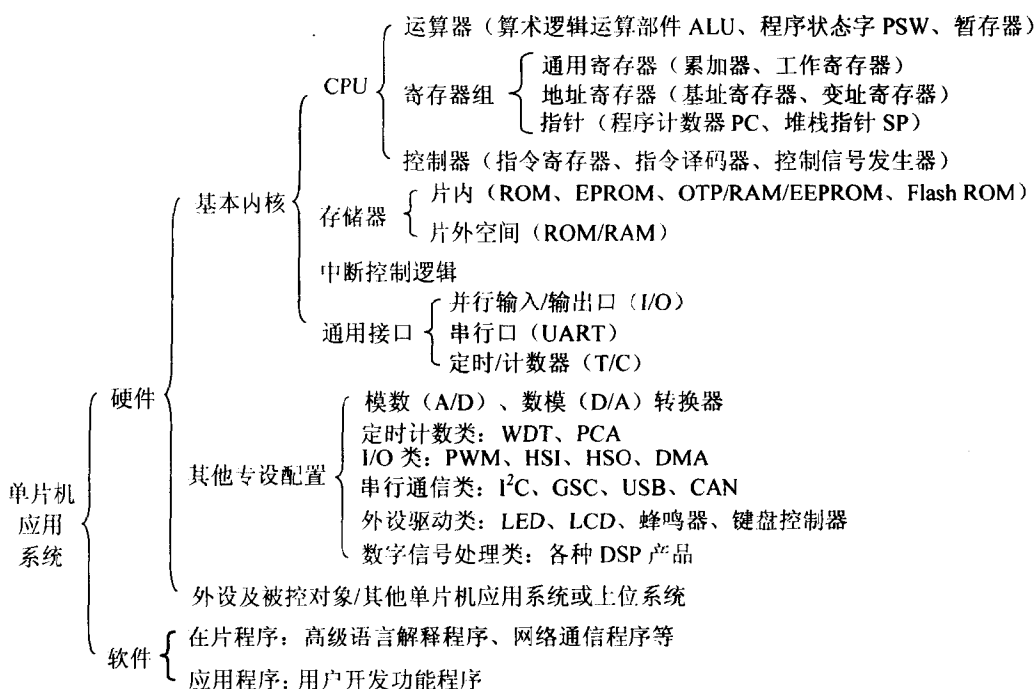


图 1.2 单片机应用系统

1. 硬件部分

• **基本内核** 结构相当于一个微型计算机, 仅由这部分组成的单片机产品是名副其实的单片微型计算机。其 CPU 往往是 4 位或 8 位的。Intel 公司的 MCS-51 单片机是 8 位机中最流行的产品, 已成为事实上的标准, 很多公司开发与其兼容的系列, 并将其作为自己高性能微控制器产品的内核。存储器一般包括两个区域, 即程序存储器区 (ROM) 和数据存储器区。为适应用户二次产品开发的需要及成本因素, 片内通常只集成部分存储器。片内程序存储器的结构又有掩模 ROM、可擦除可编程的 EPROM、一次可编程 ROM (OTP) 和无 ROM

型。有的产品中还会加入部分非易失性存储器，如 EEPROM、Flash ROM 等。

• 其他专设配置部分的各功能电路单元 可以配合前述的基本内核形式的单片机芯片另行设计多片型系统。高性能的单片机（8 位机、16 位机、32 位机）产品为针对特定的用户功能，已把它们与内核做在同一块半导体芯片里，这就是微控制器，亦即嵌入式微控制器。这些电路单元中主要有监视定时器 WDT（Watchdog Timer）、可编程计数器阵列 PCA（Programmable Counter Array）、高速输入输出 HSI 和 HSO（High-speed Input/Output）、直接存储器访问 DMA（Direct Memory Access）、串行总线 I²C（Inter-Integrated Circuit）、全局串行通道 GSC（Global Serial Channal）、USB、CAN、LED 和 LCD 驱动器、各种 DSP 等。

2. 软件部分

应用系统少不了软件——程序。因此，当设计一个单片机应用系统或产品时，尽管可以选择高性能的单片机，但软件部分仍需要自己开发。软件设计是根据指令系统及系统功能要求进行的。对一个应用系统来说，某些功能的实现既可以用硬件来完成也可以用软件来完成。用硬件实现具有速度快、占用 CPU 时间少的优点，但电路复杂、成本高；用软件来实现则可以简化硬件电路的设计、提高系统的可靠性、降低系统成本，但占用 CPU 时间，实时性差，只能用于对系统速度要求不高的场合。

此外，单片机应用系统的开发，还必须具备开发单片机应用系统的系统——开发系统。

1.2 计算机中数的表示及运算

计算机所处理的信息，必须先经过信息数字化处理，即数据、文字、符号、图像等各种信息都要经过编码，成为计算机所能识别和处理的数字信息。因此，计算机选择哪种数字系统、如何表示数据，将直接影响计算机的性能和结构。在计算机中，数字是以二进制数表示的，它易于物理实现，同时数据存储、传送和处理简单可靠，而且运算规则简单，特别是采用二进制后，能方便地使用逻辑代数这一数学工具进行逻辑电路的设计、分析、综合，并使计算机具有逻辑性。因此，有必要了解二进制表示与其他常用进位计数制间的转换方法。

1.2.1 常用数制及转换

人们习惯用十进制数，而计算机用的是二进制数。在学习和使用计算机时，为了阅读和书写的方便，常采用十六进制。

1. 各种进位计数制及其表示方法

进位计数制有两个要素：一是使用的数字符号，二是进位规则。

十进制数使用 10 个数字符号，即常用的阿拉伯符号 0、1、2、3、4、5、6、7、8、9。这些符号叫数码，数码的个数称为基数。进位规则是“逢十进一”，即计数时每一位计到 10 便往上进 1 位。所以，基数就是两相邻数码中高位的权与低位的权之比。

每个单独的数码只表示 0~9 中的一个数值。当要表示的数值较大时，需要用多个数码依次顺序排列来表示。每个数码表示的数值不仅取决于本身，还取决于它所处的位置。如 30681 表示三万零六百八十一。可写成如下多项式的形式：

$$3 \times 10^4 + 0 \times 10^3 + 6 \times 10^2 + 8 \times 10^1 + 1 \times 10^0$$

式中的 10^0 、 10^1 、 10^2 、 10^3 、 10^4 在数学上分别表示个位、十位、百位、千位、万位上的“权”。每一位上的数码与该位的“权”的积表示该位数值的大小。

任一个十进制数 N 可表示为：

$$N = \pm [a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \cdots + a_0 \times 10^0 + a_{-1} \times 10^{-1} + \cdots + a_{-m} \times 10^{-m}]$$

$$= \pm \sum_{i=-m}^{n-1} a_i \times 10^i$$

这是一个多项式。式中幂指数 m 是小数位数， n 是整数位数，均为正整数；系数 a_i 是第 i 位的数码符号，可以是 0~9 十个数码符号中的任一个，由具体数决定；10 是基数。

推广之，对于任意进位计数制，若基数用 R 表示，则任意数 N 可表示为：

$$N = \pm \sum_{i=-m}^{n-1} a_i R^i$$

对于二进制数，数码符号只有 2 个，即 0 和 1，基数是 2。进位规则是“逢二进一”。以 8 位二进制为例，从高到低各位的权依次是： 2^7 、 2^6 、 2^5 、 2^4 、 2^3 、 2^2 、 2^1 、 2^0 。如有一个二进制数 11010100，如果感觉不到它的大小，可用多项式展开法转换成十进制数，就知道它是多少了：

$$(11010100)_2 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

$$= 128 + 64 + 0 + 16 + 0 + 4 + 0 + 0 = (212)_{10}$$

对于十六进制数，数码符号需 16 个，取：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F，基数是 16。进位规则是“逢十六进一”。

作为对照，表 1.1 列出了部分十、二、十六进制数。

表 1.1 常用数制对照

十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	12	1100	C
1	0001	1	13	1101	D
2	0010	2	14	1110	E
3	0011	3	15	1111	F
4	0100	4	16	1 0000	10
5	0101	5	17	1 0001	11
6	0110	6	32	10 0000	20
7	0111	7	255	1111 1111	FF
9	1000	8	0.5	0.1	0.8
8	1001	9	0.25	0.01	0.4
10	1010	A	0.125	0.001	0.2
11	1011	B	0.0625	0.0001	0.1

2. 二进制数的运算

1) 二进制加法。二进制加法规则为：

$$0+0=0, \quad 0+1=1, \quad 1+0=1, \quad 1+1=10$$

例如，数 10101 与数 10011 相加，演算过程如下（括号内为对应十进制值）：

$$\begin{array}{r}
 10111 \quad \text{——进位} \\
 10101 \quad (21) \text{——被加数} \\
 + 10011 \quad (19) \text{——加数} \\
 \hline
 101000 \quad (40) \text{——和}
 \end{array}$$

同十进制数加法一样，两数相加，每一位要考虑三个数：两个相加的数和来自低位的进位，而结果除得到本位的和以外，还有向高位的进位。

2) 二进制减法。二进制减法规则为：

$$0-0=0, \quad 1-1=0, \quad 1-0=1, \quad 0-1=1 \text{ (有借位)}$$

例如，数 10110 减去数 1100 的过程如下：

$$\begin{array}{r}
 -10000 \quad \text{——借位} \\
 10110 \quad (22) \text{——被减数} \\
 -) 1100 \quad (12) \text{——减数} \\
 \hline
 01010 \quad (10) \text{——差}
 \end{array}$$

3) 二进制乘法。二进制乘法规则为：

$$0 \times 0 = 0, \quad 0 \times 1 = 0, \quad 1 \times 0 = 0, \quad 1 \times 1 = 1$$

例如，11011 × 1101 的运算过程如下：

$$\begin{array}{r}
 11011 \quad (27) \text{——被乘数} \\
 \times 1101 \quad (13) \text{——乘数} \\
 \hline
 11011 \quad \text{——第一次部分积} \\
 00000 \quad \text{——第二次部分积} \\
 011011 \quad \text{——第二次部分积} \\
 11011 \quad \text{——第三次部分积} \\
 10000111 \quad \text{——第三次部分积} \\
 11011 \quad \text{——第四次部分积} \\
 \hline
 101011111 \quad (351) \text{——乘积}
 \end{array}$$

4) 二进制除法。二进制除法规则为：

$$0 \div 1 = 0, \quad 1 \div 1 = 1$$

例如，1111 ÷ 101 的运算过程如下：

$$\begin{array}{r}
 \text{除数——} 101 \quad (5) \overline{) 1111 \quad (15)} \quad \text{——被除数} \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

3. 十六进制数的使用

计算机只能处理二进制数。二进制数书写起来非常麻烦，并且容易出错、不易阅读。从表 1.1 可以看出，十六进制的 16 个数码与四位二进制的 16 种状态是一一对应的。实际上，十六进制数本质上是二进制数，是二进制数的简便书写方式。例如，将二进制数

1100101011.01011 写成十六进制数的方法是，先对该数从小数点向两边 4 位分组，不足 4 位补 0；再从左至右把每一组用十六进制数码依次写出来：

$$\begin{array}{ccccccc} \underline{0011} & \underline{0010} & \underline{1011} & \underline{0101} & \underline{1000} & & \\ | & | & | & | & | & & \\ 3 & 2 & B & 5 & 8 & & \end{array}$$

为了区别，书写时，二进制数末尾加一大写的字母 B，十六进制数末尾加 H，十进制数末尾加 D（也可不加）。如上面两数的关系可以表示为：

$$1100101011.01011B = 32B.58H$$

把十六进制数展开成二进制数也很简单，只要依次将十六进制数每一位用 4 位二进制展开写出便可。例如，将 76F9.6E2H 写成二进制形式如下：

$$\begin{array}{ccccccc} & 7 & 6 & F & 9 & . & 6 & E & 2 & H \\ & \swarrow & \swarrow & \swarrow & \swarrow & & \swarrow & \swarrow & \swarrow & \\ 0111 & 0110 & 1111 & 1001 & . & 0110 & 1110 & 0010 & B \end{array}$$

一般书写时可以去掉前后的无效 0，则上述结果可表示成：

$$76F9.6E2H = 111011011111001.01101110001B$$

以上介绍的 3 种进位制，十进制是人们习惯的计数制，二进制是计算机唯一能处理的进位制，而十六进制是程序员在编写程序时，为了方便而用于代替二进制的一种数据表示方法。在使用计算机的过程中，程序员还必须熟练掌握各种数制之间的转换。

4. 数制之间的转换

当两个有理数相等时，其整数部分和小数部分一定分别相等，这是不同计数制之间转换的依据。

任意进制数之间相互转换时，整数部分和小数部分必须分别进行。

设待转换数的基数为 R_1 ，目标数的基数为 R_2 ，整数部分的转换方法是：将 R_1 进制的数的整数反复除以 R_2 ，从最低位向最高位逐次得到余数 $a_{n-1} \cdots a_0$ ，即为所求 R_2 进制数的整数部分。例如，将十进制数的整数（部分）25 转换成二进制整数和二进制数的整数（部分）101101 转换成十进制整数就是这样实现的。

小数部分的转换方法是：将 R_1 进制数的纯小数（部分）反复乘以 R_2 ，从最高位向最低位逐次进行，并取每次的整数部分，从而得到 R_2 进制小数 $0.a_{-1}a_{-2} \cdots a_{-m}$ 。例如，将十进制数的小数（部分）0.849 转换成二进制小数，将二进制数的小数（部分）0.1101 转换成十进制小数就是这样实现的。

上述四例过程如下：

$$\begin{array}{r|l} 2 & 25 \quad \text{二进制余数} \\ \hline 2 & 12 \quad \text{——} 1 \\ 2 & 6 \quad \text{——} 0 \\ 2 & 3 \quad \text{——} 0 \\ 2 & 1 \quad \text{——} 1 \\ & 0 \quad \text{——} 1 \end{array} \quad \uparrow$$

结果：25 = 11001 B

$$\begin{array}{r|l} 1010 & 101101 \quad (\text{二进制余数}) \quad \text{十进制余数} \\ \hline 1010 & 100 \quad \text{——} 101 \quad \text{——} 5 \\ & 0 \quad \text{——} 100 \quad \text{——} 4 \end{array} \quad \uparrow$$

结果：101101B = 45

积整数部分	小数 × 目标基数	十进制数码	积整数	小数 × 目标基数
	0.849×2			0.1101×1010
1	$.698 \times 2$	8 ← 1000		$.0010 \times 1010$
1	$.396 \times 2$	1 ← 0001		$.0100 \times 1010$
0	$.792 \times 2$	2 ← 0010		$.1000 \times 1010$
1	$.584 \times 2$	5 ← 0101		$.0000$
0				

取 4 位小数, 结果: $0.849 \approx 0.1101\text{B}$

结果: $0.1101\text{B} = 0.8125$

这种方法适用于两个任意进制数之间的转换, 但二进制与十六进制数之间的转换可直接使用前述的方法。

任意进制数转换为十进制数的方法, 还可用本节中的多项式展开法。

1.2.2 机器数及其编码

计算机只认识二进制数, 那么存储数据的最小单位就是一个二进制位, 这一位中只能存储“0”或“1”。在计算机中, 通常把一组相邻的二进制数码(通常为 8 位)称为字节, 它是构成信息的一个小的单位, 并作为一个整体来参加操作, 它比字小, 是构成字的单位。在计算机中, 一串数码作为一个整体来处理或运算, 称为一个计算机字, 简称字。字通常分为若干个字节。一个存储体(也通称存储器)拥有的存储单元的字节数, 叫做存储器的容量。

下面介绍在计算机中如何用二进制数表示其他进制数以及其他各种信息的。

1. 机器数与真值

数在计算机中的形式叫做机器数, 因而机器数是二进制的, 机器数又分为无符号数和带符号数两类。再考虑到计算机每次处理数据的长度(字长)和存储器单元的位数, 机器数的长度一般是 8 位的整数倍。因此, 为了简便, 这里只介绍 8 位整数的机器数。

对于 8 位无符号数, 计算机内形式是从 00000000B 到 11111111B , 十六进制为 00H 到 FFH , 即 8 位无符号机器数的表示范围为 $0 \sim 255$ 共 256 个整数。

对于带符号数——正数(符号用“+”)和负数(符号用“-”)的机器数的表示, 关键在于符号的机器表示。因为计算机中数据载体的任一位单元都是相同的电子元件, 都只能表示“0”或“1”。符号“+”和“-”也只有用一位 0 和 1 来表示, 而且占用机器数的最高位。为加深印象, 符号位暂用带底纹的数位标志。如 $+1001101$ 的机器数是 1001101 , -10101 的机器数是 10010101 。这样, 8 位带符号机器数的最高位用作符号位, 剩余 7 位用作数值位。

把这种符号“数码化”的数称做机器数, 而把原来符号没有数码化的数或书写数称为机器数的真值。例如 $+1001101$ 是 1001101 的真值, -10101 是 10010101 的真值。真值有时书写成十进制形式。

机器数解决了符号问题, 但也带来了新的问题: 计算机把符号也看成数值一起运算处理。如 $-1 + (-2)$ 应等于 -3 , 而 10000001 加上 10000010 却变成 00000011 , 即 $+3$ 。再考虑到要方便的设计计算机的运算器, 机器数还必须要进行编码, 以解决诸如此类的各种问题。

2. 机器数的编码及运算

二进制机器数的编码有 3 种: 原码、反码、补码。计算机内一般使用补码。

1) 原码 上面介绍机器数时, 简单地把数的符号“+”或“-”用符号位上的“0”或“1”代替, 数值部分即为该数本身, 这样的机器数就叫原码。

将书写数(真值)表示为原码, 一是用换符号的方法; 二是用公式: 如原数(真值)为 X , 则其原码:

$$[X]_{\text{原}} = X \quad (X \geq 0)$$

$$\text{或} \quad [X]_{\text{原}} = 2^{n-1} - X \quad (X \leq 0)$$

其中 n 为包括符号位在内的字长的位数。

例如, +5 的原码 $[+5]_{\text{原}} = 00000101\text{B}$

$$-5 \text{ 的原码 } [-5]_{\text{原}} = 2^7 - (-5) = 10000000\text{B} + 00000101\text{B} = 10000101\text{B}$$

由于原码的最高位表示符号, 则机器数 00000000B 表示“+0”, 10000000B 表示“-0”。可见, 8 位机器数原码能表示的最大正数是+127, 最小负数是-127, 而 0 则有两个。

2) 反码 对于负数, 将原码的数值位各位取反, 就得到另一种机器数的表示法, 叫反码表示法。规定正数的反码等于原码。反码也可用公式求得: 如原数(真值)为 X , 则其反码

$$[X]_{\text{反}} = X \quad (X \geq 0)$$

$$\text{或} \quad [X]_{\text{反}} = (2^n - 1) + X \quad (X \leq 0)$$

其中 n 为包括符号位在内的字长的位数。

例如, +5 的反码 $[+5]_{\text{反}} = [+5]_{\text{原}} = 00000101\text{B}$

$$\begin{aligned} -5 \text{ 的反码 } [-5]_{\text{反}} &= (2^8 - 1) + (-5) \\ &= 11111111 - 00000101 \\ &= 11111010\text{B} \end{aligned}$$

8 位反码表示范围为-127~+127, 0 也有两个: +0 (00000000B) 和 -0 (11111111B)。

3) 补码 补码表示法的产生, 是欲把负数转化成正数, 使减法变成加法, 从而使正负数的加减运算转化为单纯的正数加法运算。

“校时”问题可以帮助理解补码的概念。若现在是早上 8 点, 而手表在夜里一点就停了, 那么可以正拨 7 点, 也可以倒拨 5 点来对时。此时, +7 和 -5 是等价的。也就是说, -5 的操作可以用 +7 来实现。

规定正数的补码等于原码。负数的补码有两种求法: 一是将其反码+1, 如若 $X = -1011010$, 则

$$[X]_{\text{补}} = [X]_{\text{反}} + 1 = 10100101 + 00000001 = 10100110\text{B}$$

二是利用公式: 如原数(真值)为 X , 则其补码

$$[X]_{\text{补}} = X \quad (X \geq 0)$$

$$\text{或} \quad [X]_{\text{补}} = 2^n + X \quad (X < 0)$$

其中 n 为包括符号位在内的字长的位数。

例如, +5 的补码 $[+5]_{\text{补}} = [+5]_{\text{原}} = 00000101\text{B}$

$$\begin{aligned} -5 \text{ 的补码 } [-5]_{\text{补}} &= 2^8 + (-5) \\ &= 10000000 - 00000101 \\ &= 11111011\text{B} \end{aligned}$$

8 位补码的表示范围为-128~+127。0 只有一个, 即 00000000B。10000000B 是-128 的补码。

8 位二进制数不同编码时的值见表 1.2。

表 1.2 8 位机器数的数值编码对照表

机器数	数值编码意义			
	无符号数	原码	反码	补码
0000 0000	+0	+0	+0	+0
0000 0001	1	+1	+1	+1
⋮	⋮	⋮	⋮	⋮
0111 1111	127	+127	+127	+127
1000 0000	128	-0	-127	-128
1000 0001	129	-1	-126	-127
⋮	⋮	⋮	⋮	⋮
1111 1110	254	-126	+1	-2
1111 1111	255	-127	-0	-1

4) 补码的运算 在微型计算机的算术操作中, 都采用补码运算。故在此仅介绍补码的运算法则:

① 当 $X \geq 0$ 时, $[X]_{\text{补}} = [X]_{\text{反}} = [X]_{\text{原}}$

② $[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$

③ $[X]_{\text{补}} + [Y]_{\text{补}} = [X+Y]_{\text{补}}$

④ $[X-Y]_{\text{补}} = [X+(-Y)]_{\text{补}}$

例如, 已知 $X=48$, $Y=16$, 求 $X-Y$ 。

解: 既可用原码减法也可用补码加法, 两种方法的解题过程分别如下:

用原码做减法

$$X-Y = 48-16 = 32$$

$$\begin{array}{r} 00110000 \\ -) 00010000 \\ \hline 00100000 \end{array}$$

用补码做加法

$$\begin{aligned} X-Y &= [[X-Y]_{\text{补}}]_{\text{补}} \\ &= [[48]_{\text{补}} + [-16]_{\text{补}}]_{\text{补}} \\ &= [32]_{\text{补}} = 32 \end{aligned}$$

$$\begin{array}{r} [48]_{\text{补}}: \quad 00110000 \\ [-16]_{\text{补}}: +) 11110000 \\ \hline 100100000 \\ \swarrow \text{自然丢失} \end{array}$$

5) 运算的溢出问题 由于计算机中表示数据的字长(位数)有一定的限制, 所以数据的表示总有一个范围。如字长为 8 位时, 用补码表示带符号整数的范围是 $-128 \sim +127$ 。当两数进行算术运算时, 结果可能超出这个范围, 这便产生了溢出。因此, 计算机在做算术运算时, 必须检查溢出, 以防止发生错误。

先看几个例子:

两数相加时, 可能会产生进位, 又由于补码包括符号位和数值位两部分, 设数值位向符号位的进位为 C_s , 符号位向高位(不存在的第 9 位)的进位为 C_{s+1} 。

例 1 $98+25=123$ (未溢出)