

单片机 原理及应用

希望图书创作室 总策划
成都木马科技 编 著



本书配套光盘内容包括
与本书配套的多媒体教学软件



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



单片机 原理及应用

希望图书创作室 总策划
成都木马科技 编著



本书配套光盘内容包括
与本书配套的多媒体教学软件



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

单片机作为计算机技术的一个分支,其发展日新月异,现在,单片机的开发应用已在工业测控、机电一体化、智能仪表、家用电器、航空航天及办公自动化等各个领域占据了重要地位,本书重点介绍占当前潮流主导的 MCS-51 系列单片机。

本书由十章构成,主要内容包括:微机原理基础,单片机概述,单片机应用系统的仿真及开发,8051 单片机引脚及其功能, MCS-51 指令系统,汇编语言程序设计,定时/计数器,串行接口,键盘与显示器接口技术,传感器及模/数、数/模转换接口技术。本书从基础讲起,由浅入深地逐步涉及到高层领域。现代社会离不开网络,因此单片机中串行通信的地位越来越重要,本书中特别对单片机与单片机之间的单机、多机通信,单片机与 PC 机之间的单机通信、多机通信作了细致的论述。

本书的编者都是单片机技术开发、研究和教学工作的专家、学者,根据他们在单片机应用开发第一线所积累的实际经验编写了本书,使得本书具有内容全面、结构严谨、实例众多、语言生动、通俗易懂等特点。本书既可作为理工科大中专院校相关专业的教材,也可作为单片机爱好者或科技工作者的自学参考书。

本书配套光盘内容包括与本书配套的多媒体教学软件。

系 列 书 : “九五”国家重点电子出版物规划项目·计算机知识普及系列

书 名 : 单片机原理及应用

文 本 著 者 : 成都木马科技

责 任 编 辑 : 苏静

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

出版、发 行 者 : 北京希望电脑公司 北京希望电子出版社

地 址 : 北京海淀路 82 号, 100080

网址: www.bhp.com.cn

E-mail: lwm@hope.com.cn

电话: 010-62562329,62541992,62637101,62637102,62633308,62633309

(发行和技术支持)

010-62613322-215 (门市) 010-62531267 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心

C D 生 产 者 : 文录激光科技有限公司

文 本 印 刷 者 : 北京市媛明印刷厂

开 本 / 规 格 : 787×1092 毫米 1/16 开本 14.25 印张 311 千字

版 次 / 印 次 : 2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷

印 数 : 0001-5000 册

本 版 号 : ISBN 7-900044-00-0/TP·00

定 价 : 30.00 元 (1CD, 含配套书)

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损,本社负责调换。

前 言

单片机(也称单片微控制器)的出现是计算机发展史上的重要里程碑,它使计算机技术开始朝两个专业化方向发展,即通用计算机和微控制器两大分支。从 Intel 公司于 1976 年推出第一代单片机 MCS-48 至今,单片机技术得到迅速发展,相继推出了第二代和第三代高性能微控制器产品,单片机的开发应用已在工业测控、机电一体化、智能仪表、家用电器、航空航天及办公自动化等各个领域占据了重要地位。

由于单片机技术在各个领域正得到越来越广泛的应用,世界上许多集成电路生产厂家相继推出了各种类型的单片机。在单片机家族的众多成员中,MCS-51 系列单片机以其优越的性能、成熟的技术、高可靠性和高性价比,迅速占领了工业测控和自动化工程应用的主要市场,成为国内单片机应用领域中的主流。目前可用于 MCS-51 系列单片机开发的硬件越来越多,与其配套的各类开发系统、各类软件也越来越完善,可以极方便地利用现有资源,开发出用于不同目的的各类应用系统。因此本书重点讲述 MCS-51 系列单片机,同时详尽地讨论了与单片机系统设计相关的传感器技术、A/D 转换技术、D/A 转换技术以及单片机系统设计中必不可少的键盘、显示器和打印机接口等。现代社会离不开网络,因此单片机中串行通信的地位越来越重要,本书中特别对单片机与单片机之间的单机、多机通信,单片机与 PC 机之间的单机通信、多机通信作了细致的论述。

本书的编者都是单片机技术开发、研究和教学工作的专家、学者,根据他们在单片机应用开发第一线所积累的实际经验编写了本书,使得本书具有资料全面、结构严谨、实例众多、语言生动、通俗易懂等特点。本书可作为理工科大中专院校的教材,也可作为单片机爱好者或科技工作者的参考资料。

成都木马科技有限公司
1999 年 10 月

JSP03/06



目 录

第一章 微机原理基础 1	第三章 单片机应用系统的仿真及开发 . 57
第一节 计算机中的数制与逻辑..... 1	第一节 单片机应用系统的设计..... 57
一、常用数制..... 1	一、应用系统的硬件设计..... 57
二、常用数制之间的转换..... 2	二、应用系统的软件设计..... 58
三、二进制数运算..... 4	第二节 单片机应用系统的开发..... 58
四、常用逻辑运算..... 7	第四章 8051 单片机引脚及其功能 60
第二节 常用芯片..... 9	第一节 引脚概述..... 60
一、集成 TTL 基本门电路及常用芯片..... 9	第二节 输入/输出端口结构..... 62
二、三态门..... 11	一、P0 口..... 62
三、译码器..... 12	二、P1 口..... 64
四、地址锁存器..... 14	三、P2 口..... 64
五、采样及控制开关..... 15	四、P3 口..... 65
七、半导体存储器..... 20	五、端口的负载能力和接口要求..... 66
第三节 微机的组成及工作过程..... 26	第三节 扩展程序存储器..... 67
一、数字电子计算机的基本结构..... 26	一、扩展总线..... 67
二、计算机软件..... 28	二、扩展 8K 字节 EPROM..... 68
三、微型计算机概述..... 29	三、扩展 16K 字节 EPROM..... 69
第二章 单片机概论 34	第四节 扩展数据存储器..... 70
第一节 单片机发展史及应用..... 34	一、扩展总线..... 70
一、概述..... 34	二、8051 扩展 2K 字节 RAM..... 70
二、单片机发展概况..... 34	第五节 程序存储器与数据存储器同时扩展... 70
三、单片机主流产品系列..... 34	一、8031 外扩 32K EPROM 和 32K RAM... 70
四、单片机的应用领域..... 35	二、译码法扩展大容量存储器..... 72
第二节 MSC-51 综述..... 36	第六节 并行 I/O 口的直接应用..... 74
一、系统资源..... 36	一、I/O 口的直接输入/输出..... 74
二、8051 存储器配置..... 37	二、开关电路及驱动电路接口..... 77
三、复位..... 44	三、外部 I/O 的扩展..... 80
四、工作原理(CPU 时序和其他电路)..... 45	第五章 MCS-51 指令系统 84
五、中断系统..... 50	第一节 指令系统综述..... 84



一、指令格式.....	84	二、条件转移指令.....	102
二、指令描述中的符号.....	85	三、子程序调用及返回指令.....	103
三、指令分类.....	85	四、空操作指令.....	104
第二节 寻址方式.....	85	第七节 布尔操作类指令.....	105
一、寄存器寻址.....	86	一、位传送指令.....	105
二、直接寻址.....	86	二、位置位/复位指令.....	105
三、寄存器间接寻址.....	86	三、位运算指令.....	105
四、立即寻址.....	87	四、位控制转移指令.....	106
五、基址寄存器加变址寄存器间接寻址.....	87	第六章 汇编语言程序设计.....	108
六、相对寻址.....	88	第一节 汇编语言源程序的格式.....	108
七、位寻址.....	88	一、标号.....	109
第三节 数据传送类指令.....	89	二、操作数.....	109
一、内部 RAM 的数据传送指令.....	89	第二节 伪指令.....	109
二、外部 RAM 数据传送指令.....	90	一、ORG (汇编起始命令).....	110
三、程序存储器取数据指令.....	91	二、END (汇编结束命令).....	110
四、数据交换指令.....	91	三、EQU (等值命令).....	110
五、堆栈操作指令.....	92	四、DATA (数据地址赋值命令).....	110
第四节 算术运算类指令.....	93	五、DB (定义字节指令).....	111
一、加法指令.....	93	六、DW (定义字命令).....	111
二、带进位加法指令.....	94	七、DS (定义空间命令).....	112
三、加 1 指令.....	94	八、BIT (位地址符号命令).....	112
四、减法指令.....	94	第三节 汇编语言源程序的人工汇编.....	112
五、减 1 指令.....	95	第四节 MCS-51 程序设计举例.....	114
六、乘除指令.....	95	一、简单程序.....	114
七、十进制调整指令.....	96	二、分支程序.....	116
八、算术运算指令使用举例.....	96	三、循环程序.....	122
第五节 逻辑操作类指令.....	98	四、查表程序.....	125
一、逻辑“与”指令.....	98	五、子程序.....	126
二、逻辑“或”指令.....	98	第七章 定时/计数器.....	130
三、逻辑“异或”指令.....	98	第一节 定时/计数器概述.....	130
四、累加器清零、取反指令.....	99	第二节 定时器的控制字.....	131
五、移位指令.....	99	一、工作模式寄存器 TMOD(89H).....	131
六、逻辑指令使用举例.....	99	二、控制寄存器 TCON(88H).....	132
第六节 控制转移类指令.....	100	第三节 定时/计数器的四种工作模式.....	133
一、无条件转移指令.....	100	一、模式 0.....	133



二、模式 1.....	134	五、主机查询, 从机中断方式的多机通信	
三、模式 2.....	134	软件实例.....	170
第四节 定时器/计数器的编程和应用.....	136	六、主机、从机构成中断方式的多机通信	
一、模式 0 的应用.....	136	软件实例.....	173
二、模式 1 的应用.....	137	第五节 PC 机与单片机的通信技术基础.....	177
三、模式 2 的应用.....	137	一、异步通信芯片 8250 概述.....	179
四、模式 3 的应用.....	138	二、8250 的内部寄存器.....	182
五、运行中读定时器/计数器.....	139	三、8250 的初始化.....	186
六、门控制位 GATE 的功能和使用方法.....	139	第六节 PC 机与 8031 双机通信技术.....	187
第八章 串行接口.....	141	一、通信接口设计.....	187
第一节 串行通信的基本知识.....	141	二、通信软件设计.....	188
一、数据通信的概念.....	141	第七节 PC 机与 8031 多机通信技术.....	192
二、串行通信的传送方向.....	141	一、多机通信原理.....	193
三、异步通信和同步通信.....	142	二、通信接口设计.....	194
四、波特率(Baud rate).....	143	三、通信软件设计.....	194
第二节 8051 串行接口.....	144	第九章 键盘与显示器接口技术.....	201
一、结构.....	144	第一节 键盘接口技术.....	201
二、串行口控制字及控制寄存器.....	145	一、矩阵键盘工作原理.....	202
三、串行通信工作方式.....	147	二、按键的识别方法.....	202
四、波特率设计.....	150	三、键盘的编码.....	204
第三节 单片机双机通信技术.....	153	第二节 LED 显示器接口技术.....	204
一、通信接口设计.....	153	一、显示器结构原理.....	204
二、查询方式双机通信软件设计.....	154	二、LED 显示器接口及显示方式.....	205
三、中断方式双机通信软件设计.....	158	第十章 传感器及模/数、数/模转换	
四、通信软件的调试及故障排除.....	162	接口技术.....	207
第四节 单片机多机通信技术.....	162	第一节 传感器.....	207
一、多机通信接口设计.....	162	第二节 数/模、模/数转换技术.....	208
二、多机通信原理.....	163	一、D/A 转换器接口技术.....	208
三、多机通信的软件协议.....	163	二、A/D 转换器接口技术.....	217
四、查询方式多机通信软件实例.....	164		

第一章 微机原理基础

本章主要讲述微机基本常识，包括：常用数制、逻辑运算、常用芯片等。

第一节 计算机中的数制与逻辑

我们在日常生活中最熟悉的数制是十进制，但计算机的电状态只有两种，即通和断，因而在计算机内部只能使用二进制，计算机内部使用的这些二进制信息在不同的场合可能代表不同的意义，整个计算机系统的工作就是对这些二进制信息进行存储、传送、运算和逻辑判断。但人与计算机之间的交流，如用键盘输入程序、数据，或用数码管、CRT 显示器显示信息，以及用打印机打印数据等使用的多是十进制或十六进制，因此，我们必须学习不同数制及其转换。

一、常用数制

(一) 十进制数 (可用 D 作后缀，一般省略不写)

十进制用 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 十个数字来表示数，十进制的基数是 10，计数时，每一位计到十就往上进一位，也就是逢十进一，或者说上一位数是下一位的十倍。

如果 X_i 表示 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 十个数中任意一个，那么，一个含有 n 位整数， m 位小数的十进制数可表示为：

$$N = X_{n-1} \times 10^{n-1} + X_{n-2} \times 10^{n-2} + \dots + X_0 \times 10^0 + X_{-1} \times 10^{-1} + \dots + X_{-m} \times 10^{-m}$$

或写成：

$$N = \sum_{i=-m}^{n-1} X_i \times 10^i$$

其中， N 表示该数的十进制值。

(二) 二进制数 (用 B 作后缀)

以 2 为基数的数制叫二进制，它只包括两个符号，即 0 和 1，在一个二进制数中，高一位的权是低一位的两倍，对于整数，从右往左各位的权依次是 1, 2, 4, 8, 16...对于小数，从左往右各位的权分别是 1/2, 1/4, 1/8, 1/16...把十进制表示式中的 10 换成 2 就得到二进制表示式。如果 X_i 表示 0, 1 两个数中的任一个，那么一个含有 n 位整数， m 位小数的二进制数可表示为：

$$N = X_{n-1} \times 2^{n-1} + X_{n-2} \times 2^{n-2} + \dots + X_0 \times 2^0 + X_{-1} \times 2^{-1} + \dots + X_{-m} \times 2^{-m}$$

或写成：

$$N = \sum_{i=-m}^{n-1} X_i \times 2^i$$

其中, X_i 即为第 i 位的系数, 2^i 为第 i 位的权。

如: 十进制的 5 可表示成二进制: 101B

十进制的 0.75 可表示成二进制 0.11B

十进制的 5.75 可表示成二进制 101.11B

(三) 十六进制 (用 H 作后缀)

尽管计算机内部采用二进制来表示信息, 但为了方便书写和阅读, 我们经常采用十六进制, 即在计数时逢十六进一, 这样书写的长度很短, 且十六进制数与二进制数之间的转换相当方便, 四位二进制刚好对应一位十六进制。

十六进制用 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 代表的数值与它们在十进制中的大小一样, 而 A 的数值为 10, B 为 11, C 为 12, D 为 13, E 为 14, F 为 15。

如果 X 表示 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 十六个数中的任何一个, 那么, 一个含有 n 位整数, m 位小数的十六进制数可表示为:

$$N = X_{n-1} \times 16^{n-1} + X_{n-2} \times 16^{n-2} + \dots + X_0 \times 16^0 + X_{-1} \times 16^{-1} + \dots + X_{-m} \times 16^{-m}$$

或写成:

$$N = \sum_{i=-m}^{n-1} X_i \times 16^i$$

其中, X_i 即为第 i 位的系数, 16^i 为第 i 位的权, N 为该十六进制数的十进值。

如: 十进制的 5 表示成十六进制的 5H, 十进制的 12 表示成十六进制的 0CH (注意: 在十六进制中, 当某数的最高位为 A, B, C, D, E, F 中的某一个时, 必须在最高位前添 0, 否则单片机汇编语言将其当作标号处理)。

十进制的 35 表示十六进制的 23H。

二、常用数制之间的转换

(一) 二进制 \rightarrow 十进制

公式为:

$$N = \sum_{i=-m}^{n-1} X_i \times 2^i$$

如: $101B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$

$0.11B = 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 0.75$

$101.11B = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 5.75$

(二) 十六进制 \rightarrow 十进制

公式为:

$$N = \sum_{i=-m}^{n-1} X_i \times 16^i$$

如： $0CH=0\times 16^1+12\times 16^0=12$
 $11H=1\times 16^1+1\times 16^0=17$
 $1A3H=1\times 16^2+10\times 16^1+3\times 16^0=269$

(三) 十进制→二进制

把一个十进制整数依次除以 2，并记下每次所得的余数（余数总是 1 或 0），最后所得的余数的组合即为相应的二进制数。第一位余数为最低位（LSB），最后一个余数为最高位（MSB）。

如：将十进制数 37 转换成二进制数（如右）：

其转换成的二进制数为 100101B。

同样，要将一个十进制小数转换成二进制，则应将十进制小数连续不断地乘以 2，并且记录所得的溢出数（即整数部分），直到乘积为零为止。但有时结果永不为 0，此时，只要转换到所要求的精度为止即可，第一次溢出的数为小数的最高位，最后一次溢出的为最低位。如：要将十进制数 0.3125 转换成二进制

$$\begin{array}{r|l}
 \text{LSB } 1 & 37 \\
 0 & 18 \\
 1 & 9 \\
 0 & 4 \\
 0 & 2 \\
 \text{MSB} & 1
 \end{array}$$

$0.3125\times 2=0.625$; 小数部分 0.625 溢出 0←MSB
 $0.625\times 2=1.25$; 小数部分 0.25 溢出 1
 $0.25\times 2=0.5$; 小数部分 0.5 溢出 0
 $0.5\times 2=1.0$; 小数部分 0 溢出 1←LSB

所以转换成的二进制数就为 0.0101B。

如果十进制数包含整数和小数两部分，则必须将整数和小数分别进行转换。

如：将 19.375 转换成二进制数

① 先转换 19：

$$\begin{array}{r|l}
 1 & 19 \\
 1 & 9 \\
 0 & 4 \\
 0 & 2 \\
 & 1
 \end{array}
 \quad , \begin{array}{l}
 19 \text{ 除以 } 2 \text{ 商 } 9 \text{ 余 } 1 \\
 9 \text{ 除以 } 2 \text{ 商余 } 1 \\
 \text{除以 } 2 \text{ 商余 } 0 \\
 2 \text{ 除以 } 2 \text{ 商 } 1 \text{ 余 } 0
 \end{array}$$

即 19 的二进制为 10011B；

② 再转换 0.375：

$0.375\times 2=0.75$; 小数部分为 0.75，溢出为 0
 $0.75\times 2=1.5$; 小数部分为 0.5，溢出为 1
 $0.5\times 2=1.0$; 小数部分为 0，溢出为 1

即 0.375 的二进制为 0.011B。

所以 19.375 的二进制为 10011.011B。

(四) 十六进制→二进制

十六进制的每位都与四位二进制相对应，要将十六进制转换成二进制，只要将每位十六进制数转换成相应的四位二进制数。

如将 3A.4BH 转换成二进制为:

00111010.01001011B 即 111010.01001011B

注意 此例中转换结果的二进制最高两位 0 可以省去。但在转换过程中一位十六进制一定对应四位二进制, 在这个过程中, 0 都不能省略, 否则会出错。

(五) 二进制→十六进制

由于十六进制的每一位都与四位二进制数相对应, 要将二进制数转换成十六进制数时, 若二进制数的小数点左边位数不为 4 的整数倍, 则应在最高位前添 0 使其左边位数为 4 的整数倍, 若小数点后的位数不为 4 的整数倍, 也应在小数最低位后添加 0, 使其位数为 4 的整数倍, 然后将每 4 位一组转换成相应的十六进制。

例: $11010100B = 11010100B = 0C4H$

$110010B = 00110010B = 32H$

$0.101B = 0.1010B = 0.AH$

$110.001B = 0110.0010B = 6.2H$

(六) 十进制→十六进制

十进制转换成十六进制与十进制转换成二进制方法一样, 只是除数为 16 而不是 2。

例: 将 156 转换成十六进制:

$$\begin{array}{r} C \overline{)156} \\ \underline{9} \\ 9 \end{array} \quad 156 \text{ 除以 } 16 \text{ 商 } C(\text{即 } 12) \text{ 余 } 9$$

即 156 转换成十六进制为 9CH

三、二进制数运算

(一) 二进制加法

二进制加法与十进制加法类似。比如, 两个十进制数 1968 与 1231 相加, 和为 3199。相加过程为:

$$\begin{array}{r} \text{进位:} \quad 0100 \\ \text{加数:} \quad 1968 \\ \text{被加数} + 1231 \\ \hline \text{和:} \quad 3199 \end{array}$$

将最低位的 8 和 1 相加, 得到和为 9, 表示成和为 9, 进位为 0, 然后进位被加到前一位; 将前一位的 6 和 3 以及最低位的进位相加得到和 9, 无进位; 如此继续下去直到所有列 (包括进位) 都加完为止最后得到和为 3199。

当两个二进制数相加时, 运算方法是一样的, 二进制相加的规则如下:

- ① $0+0=0$
- ② $0+1=1$
- ③ $1+1=0$, 进位 1
- ④ $1+1+1=1$, 进位 1

下面我们看一下二进制的加法过程, 四位二进制数 1101 与 1101 相加:

$$\begin{array}{r}
 \text{进位:} \quad 1101 \\
 \text{加数:} \quad 1101 \\
 \text{被加数} + 1101 \\
 \hline
 \text{和:} \quad 11010
 \end{array}$$

在第一列(最后列), 1加1等于0并向第二列进位1; 第二列, 0加0再加上第一列的进位1等于1, 无进位; 第三列, 1加1等于0, 进位1; 第四列, 1加1再加上第三列的进位1等于1, 进位, 产生第五位1。这样就得到和为11010B。

让我们再看一下8位二进制的加法, 10001111B与10110101B相加:

$$\begin{array}{r}
 \text{进位:} \quad 10111111 \\
 \text{加数:} \quad 10110101 \\
 \text{被加数} + 10001111 \\
 \hline
 \text{和:} \quad 101000100
 \end{array}$$

在单片机运算过程中, 两个8位二进制数相加的和向高位有进位时, 设标记为Cy(进位标志)=1; 如无进位, 标记为Cy=0。如果低四位(D3)向高位(D4)有进位, 标记为AC(半进位标志)=1; 如无进位, 标记为AC=0。

(二) 二进制减法

二进制减法与十进制减法类似, 我们先看一下十进制减法, 8303减去5486, 差为2817:

$$\begin{array}{r}
 \text{借位以后的被减数:} \quad 712913 \\
 \text{被减数:} \quad 8303 \\
 \text{减数:} \quad -5486 \\
 \hline
 \text{差:} \quad 2817
 \end{array}$$

因为最低位的减数6大于被减数3, 所以从被减数的高一位中借一个1, 如果此数为0(如本例); 则再从高一位不为0的数中借1, 该数减少1(本例中从3减至2), 被减数中跳过的0值位成为9, 如本例; 这等于从30中取走1, 得结果为29。在十进制中, 借位的数值为10, 所以现在被减数的值是13, 13减6等于7; 第二列9减8等于1; 第三列中, 由于减数大于被减数, 所以从高一位中借1, 使被减数由2增加到12, 12减4等于8; 第四列, 由于前面的借位, 被减数由8减至7, 7减5等于2。

当被减数的当前操作位值小于减数的对应位值时, 总是从较高一位数中借1, 该借位在数值上等于这种数制的基数。因此, 在十进制数中借一位等于10, 而在二进制中借一位等于2。

二进制减法规则如下:

- ① $0-0=0$
- ② $1-1=0$
- ③ $1-0=1$
- ④ $0-1=1$, 借位1

下面举例说明二进制减法, 11011减去1101。

$$\begin{array}{r}
 \text{借位以后的被减数:} \quad 01011
 \end{array}$$

$$\begin{array}{r}
 \text{被减数:} \qquad \qquad \qquad 11011 \\
 \text{减数:} \qquad \qquad \qquad \quad - \quad 1101 \\
 \hline
 \text{差:} \qquad \qquad \qquad \qquad \quad 1110
 \end{array}$$

“借位以后的被减数”是指产生借位以后每位被减数的值。记住，二进制的 10 相当于十进制的 2。

第一列中，1 减 1 等于 0；第二列 1 减 0 等于 1；第三列 0 减 1 需从第四位借位，于是从 10B 减 1 等于 1；由于前面的借位，第四列的被减数现在是 0，所以要从第五位借位，因此在第四列 10B 减 1 等于 1；由于前面的借位，第五列的被减数现在是 0，而减数也是 0（不存在），0 减 0 等于 0，第五列的 0 在差中不表示（因为它不是有效位）。于是 11011B 减去 1101B 后的差是 1110B。下面再看一例：01000100 减去 10100010。

$$\begin{array}{r}
 \text{借位以后的被减数:} \quad 1001000010 \quad 0 \\
 \text{被减数:} \qquad \qquad \qquad \quad 010001 \quad 0 \quad 0 \\
 \text{减数:} \qquad \qquad \qquad \quad - \quad 101000 \quad 1 \quad 0 \\
 \hline
 \text{差:} \qquad \qquad \qquad \quad 1 \quad 101000 \quad 1 \quad 0
 \end{array}$$

上述 8 位二进制减法，被减数小于减数，产生借位，在单片机中标记为 $Cy=1$ ；如果够减，不产生借位，记为 $Cy=0$ 。

注意 此时差的二进制最高位为 1，并不是数越减越大了，而表示此数为负。

（三）二进制乘法

乘法运算可看作是被乘数自身多次相加，相加的次数由乘数的数值决定。

例如：324 乘以 223

$$\begin{array}{r}
 \text{被乘数:} \qquad \qquad \qquad \quad 3 \quad 2 \quad 4 \\
 \text{乘数:} \qquad \qquad \qquad \quad \times 2 \quad 2 \quad 3 \\
 \hline
 \text{第一部分积:} \qquad \qquad \quad 9 \quad 7 \quad 2 \\
 \text{第二部分积:} \qquad \qquad \quad 6 \quad 4 \quad 8 \\
 \text{第三部分积:} \qquad \quad 6 \quad 4 \quad 8 \\
 \hline
 \text{进位:} \qquad \qquad \quad 0 \quad 1 \quad 2 \quad 1 \\
 \hline
 \text{最终积:} \qquad \qquad \quad 7 \quad 2 \quad 2 \quad 5 \quad 2
 \end{array}$$

这种乘法是把乘数与乘数的每一位相乘，然后把各部分积加起来得到了最终积。请注意，为了方便起见，加法的进位数放在部分积的下面，而不像正规加法中的放在其上面。与十进制类似，二进制乘法规则如下：

$$\begin{array}{ll}
 0 \times 0 = 0 & 0 \times 1 = 0 \\
 1 \times 0 = 0 & 1 \times 1 = 1
 \end{array}$$

请看 1111 与 1101 相乘的过程。

$$\begin{array}{r}
 \text{被乘数:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \\
 \text{乘数:} \qquad \qquad \qquad \times \ 1 \ 1 \ 0 \ 1 \\
 \hline
 \text{第一部分积:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \\
 \text{第二部分积:} \qquad \qquad \qquad 0 \ 0 \ 0 \ 0 \\
 \hline
 \text{进位:} \qquad \qquad \qquad 0 \ 0 \ 0 \ 0 \\
 \hline
 \text{部分积之和} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \\
 \text{第三部分积:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \\
 \hline
 \text{进位:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \ 0 \ 0 \\
 \hline
 \text{部分积之和:} \qquad \qquad \qquad 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \\
 \text{第四部分积之和:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \\
 \hline
 \text{进位:} \qquad \qquad \qquad 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \\
 \hline
 \text{最终乘积:} \qquad \qquad \qquad 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1
 \end{array}$$

(四) 二进制除法

除法是乘法的逆运算，它是确定一个数可以从另一个数中减去多少次的过程。我们先看一下十进制的除法。

$$\begin{array}{r}
 \text{除数 } 45 \quad \begin{array}{r} 004 \\ \hline 181 \\ \hline 180 \\ \hline 1 \end{array} \quad \begin{array}{l} \text{商} \\ \text{被除数} \\ \\ \text{余数} \end{array}
 \end{array}$$

二进制除法可仿照以上过程，比如 100011 除以 101：

$$\begin{array}{r}
 \text{除数 } 101 \quad \begin{array}{r} 000111 \\ \hline 100011 \\ \hline 101 \\ \hline 111 \\ \hline 101 \\ \hline 101 \\ \hline 101 \\ \hline 0 \end{array} \quad \begin{array}{l} \text{商} \\ \text{被除数} \\ \\ \text{余数} \\ \\ \text{余数} \\ \\ \text{余数} \\ \\ \text{余数} \end{array}
 \end{array}$$

四、常用逻辑运算

二极管和三极管工作在开关状态时，它的“导通”与“截止”可代表开关的“开通”与“关断”，开与关这两种状态可用二元常量 0 和 1 来表示。数字电路的输入、输出量一般用高、低电位（或高、低电平）来表示。高低电位亦可用二元常量来表示。那么，什么是高、低电平呢？所谓高电平就是指某一点的电位高于某一指定值，用逻辑符号“1”表示，所谓低电平就是指某一点的电位低于某一指定值，用逻辑符号“0”表示。

就其整体而言，数字电路的输出量与输入量之间的关系是一种因果关系，它可用逻辑关系来描述，因而数字电路又称为逻辑电路。

在数字电路中，所谓“门”就是实现一些基本逻辑关系的电路。最基本的逻辑关系可

归纳为与、或、非三种。所以最基本的逻辑门为与门、或门和非门。

下面介绍单片机常用外围集成电路芯片知识。对于三种最基本的逻辑关系，存在着三种基本逻辑电路。

(一) “与”逻辑关系及与门电路

当决定一件事情的各个条件全部具备时，这件事才会发生，这样的因果关系我们称之为“与”逻辑关系。

实际生活中，这种“与”的关系比比皆是。如在图 1.1 中，只有当开关 A 与 B 都闭合时，灯 Z 才会亮。所以，对灯亮来说，开关 A、B 闭合是“与”逻辑关系，并记为 $Z=AB$ ，读成 Z 等于 A 与 B。

实现“与”逻辑关系的电路称为与门，符号见图 1.2。

表达式为： $Z=AB$ ，真值表见表 1-1。

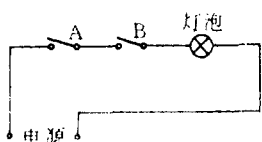


图 1.1



图 1.2

表 1-1 与门真值表

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

(二) “或”逻辑关系及或门电路

当决定一件事情的各个条件中，只要具备一个或一个以上条件，这件事情就会发生，这样的因果关系称之为“或”逻辑关系。如图 1.3 所示电路中，对灯 Z 亮来说，开关 A、B 闭合是“或”逻辑关系。因为 A 或者 B 只要有一个闭合，灯就会亮，记作 $Z=A+B$ ，读成 Z 等于 A 或 B。

实现“或”逻辑关系的电路称为或门。符号如图 1.4 所示，图中 A、B 为输入信号，Z 为输出信号。

或门真值表见表 1-2。

其表达式为： $Z=A+B$ 。

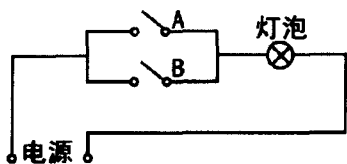


图 1.3

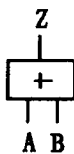


图 1.4

表 1-2 或门真值表

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

(三) “非”逻辑关系及非门（反相器）

非就是反。例如在图 1.5 中，开关 A 闭合与灯 Z 亮是“非”的逻辑关系。因为当 A 闭合时，灯灭。A 断开时，灯亮。并记作 $Z=\bar{A}$ ，读作 Z 等于 A 非或 A 反。

(四) 与非门和或非门

三种基本逻辑关系与、或、非之间还可以相互组合形成较复杂的一些逻辑关系。与门和非门连接起来就可以组成与非门，或门与非门连接起来可以形成或非门。

与非门的真值表如表 1-3 所示，逻辑符号如图 1.6 所示，其表达式为： $Z = \overline{AB}$ 。

或非门的真值表如表 1-4 所示，逻辑符号如图 1.7 所示，其表达式为： $Z = \overline{A+B}$ 。

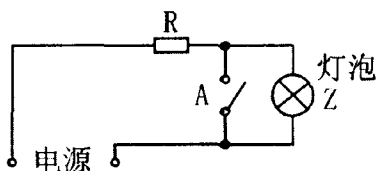


图 1.5

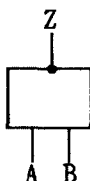


图 1.6

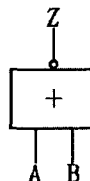


图 1.7

表 1-3 与非门真值表

A	B	Z
1	1	0
0	1	1
1	0	1
0	0	1

表 1-4 或非门真值表

A	B	Z
1	1	0
0	1	0
1	0	0
0	0	1

第二节 常用芯片

一、集成 TTL 基本门电路及常用芯片

TTL (Transistor-Transistor Logic) 是晶体管-晶体管逻辑集成电路，是一种单片集成电路。在这种集成电路中，一个逻辑电路中的所有元件和连线，都制作在同一块半导体基片上。由于这种数字集成电路的输入端和输出端电路的结构形式都采用了半导体三极管，所以一般称为晶体管。晶体管逻辑电路，简称 TTL 电路。这种集成电路比前面介绍的由二极管和三极管构成的 DTL 电路工作速度快，是实用的集成电路芯片。

在 TTL 电路中，常规定高电平额定值为 3V，低电平为 0.2V，但也不是一个固定不变的值，通常认为从 0V 到 0.8V 都算作低电平，从 2V 到 5V 都算作高电平。

下面我们介绍一些常用的实际 TTL 集成电路芯片。

(一) 74LS04 (六非门)

7404 芯片内部有六个独立的非门电路，其内部逻辑如图 1.8 所示。

逻辑表达式为： $Y = \overline{A}$ 。

(二) 74LS00 (二输入端四与非门)

74LS00 内有四个独立的二输入端的与非门电路。片内逻辑图及引脚如图 1.9。逻辑关系表达式为： $Y = \overline{A \cdot B}$

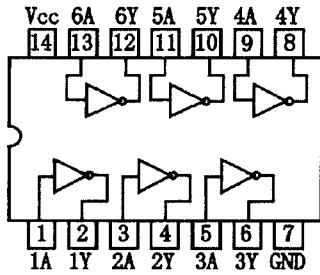


图 1.8

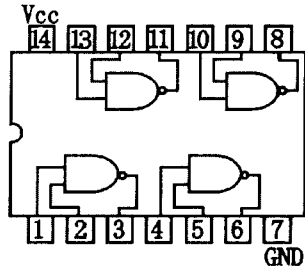


图 1.9

7410是内部含有三个独立的三输入端与非门的芯片；7411是高速三输入端三个独立的与非门芯片。

74LS30 是八输入端与非门芯片。74LS30 的逻辑图及引脚如图 1.10 所示。其逻辑表达式为： $Y = \overline{A \cdot B \cdot C \cdot D \cdot E \cdot F \cdot G \cdot H}$

(三) TTL 或非门

74LS02 内部含有四个独立的二个输入端的或非 9 门电路，引脚如图 1.11 所示。其逻辑关系表达式为： $Y = \overline{A+B}$ 。

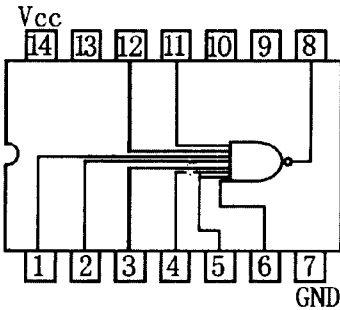


图 1.10

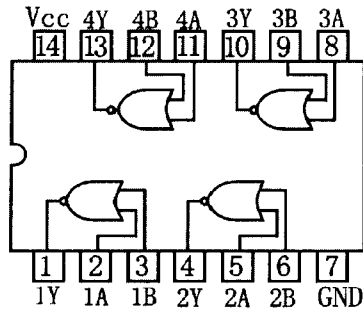


图 1.11

7427 是含有三个独立的三输入端正或非门芯片。

(四) 集电极开路门 (OC 门)

图 1.12 表示一种 OC 门的内部结构。

OC 门的输出与输入逻辑关系与普通的 TTL 与非门相同，不同的是 OC 门的输出管的集电极是开路的，使用时需外接集电极电阻 R_c 代替 TTL 与非门中复合管 (T3T4) 组成的有源负载。

当几个 OC 的输出端相连时，一般可共用一个负载电阻 R_c ，如图 1.13 所示。

常用 OC 门有 74LS06、07 等。

74LS06 引脚如图 1.14 所示。

7406 有六个独立的集电极开路输出的反向驱动器，它的输出端可加 30V 高压，可带 40mA 的负载。

74LS07 也是 OC 门，它包含 6 个独立的同相驱动器，输出端可加 30V 高压，可带 40mA 负载。