

玩转
单片机

51单片机

非常入门 与视频演练

刘建清 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

玩转单片机

51 单片机非常入门与 视频演练

刘建清 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书简要介绍了 51 单片机的基本知识、实验方法和指令系统,并演练了大量适宜初学者入门的典型实例;为方便读者实验,作者为本书的所有实例开发了 DD-900mini 实验板,并以视频的方式记录了书中主要实验的演示过程和现象。尤其珍贵的是实例中的多数子程序均具有较强的通用性,读者只需将其简单修改甚至不用修改,即可移植到自己开发的产品中。

全书语言通俗、实例丰富、图文结合、简单明了,适合单片机爱好者和单片机初学者学习,也可作为中等专业技术学校、中等职业学校等教学用书。

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

51 单片机非常入门与视频演练 / 刘建清编著. —北京:电子工业出版社, 2010.5
(玩转单片机)
ISBN 978-7-121-10774-0

I. ①5… II. ①刘… III. ①单片微型计算机 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2010) 第 075989 号

责任编辑:徐 静 康 霞 特约编辑:孙志明

印 刷:北京智力达印刷有限公司

装 订:北京中新伟业印刷有限公司

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:787×1092 1/16 印张:12.5 字数:312 千字

印 次:2010 年 5 月第 1 次印刷

印 数:4000 册 定价:42.00 元(含光盘 1 张)

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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

单片机就是把一个计算机系统集成到一个芯片上，简单地讲，一块芯片就成了一台计算机。目前市面上流行的单片机，其价格出奇地便宜。对于广大单片机爱好者来说，真是上帝的礼物。只要你玩起了单片机，你就会有一种成就感，我怎么这样聪明！单片机，再结合适当的硬件接口电路，有什么事情做不到呢？我对它的评价是八个字：软硬兼施，老少皆宜。

单片机虽然好玩，但是，很多人经过一番探索之后却深感学好单片机并非易事，甚至连入门都感到困难。作者本人也是从一位电子爱好者成长为一名电子工程师的，此过程自然少不了学习、探索、实践、再学习、再实践这样一条规律。因此，深切地知道学习单片机难，主要是不得要领，难以入门。一旦找到学习的捷径，入了门，掌握简单程序的编写方法并观察到实际演示效果，那么，必然信心大增。接下来，再向深度、广度进军时，心里就比较坦然了，最终能够一步一个脚印地去扩展自己的知识面，成为单片机的编程高手。

在与众多的单片机爱好者交流中得知，单纯讲单片机内部结构、指令太枯燥，且不易理解。他们感兴趣的是单片机编程的应用实例，而且主要喜欢简单、实用、有趣的初级实例。因此，编写本书的思路是以实战演练为主线贯穿全书，且多数实例采用视频的方式进行演示。这样，初学者能够看得清、听得到、学得快，从而达到很好的立体学习效果。

在内容安排上，本书通过 51 单片机内部资源（中断系统、定时/计数器、串口通信）、键盘接口、LED 数码管显示、LCD 液晶显示、DS1302 时钟芯片、I²C 总线接口芯片 AT24C04、DS18B20 温度传感器、红外遥控、音乐发声等大量具体的实际例子，系统演练了 51 单片机中最为常用、最为典型的接口应用。另外，本书也包括了一些作者在学习和实际设计过程中总结的经验及方法，希望能够帮助读者更好地学习 51 单片机。

本书安排的例子大部分是由作者编写的，有一些是参考相关资料改写的，全部程序都由作者调试并通过。对于例子的使用说明也尽量详细，力争让读者“看则能用，用则能成”，保证读者在动手的过程中常常体会到成功的乐趣。另外，书中的所有实例，都是基于作者设计的 DD-900mini 实验板之上的。本书附带的光盘中含有所有实验的完整源程序、视频演示和工具软件。

本书主要面向的读者是具有一定的 51 单片机基础或刚接触 51 单片机的电子爱好者，对于已经熟悉 51 单片机开发的工程师则意义不大。

本书编写过程中，参阅了《无线电》、《单片机与嵌入式系统应用》等杂志，并从互联网上搜索了一些有价值的资料，由于其中的很多资料经过多次转载，已经很难查到原始出处，在此谨向资料提供者表示感谢。

参与本书编写的人员有刘建清、贾绪岩、李凤伟、陈素侠、孙保书、刘为国等，最后由中国电子学会高级会员刘建清先生组织定稿。由于编著者水平有限，加之时间仓促，书中难免会有疏漏和不足之处，恳请专家和读者不吝赐教。

如果您在使用本书的过程中有任何问题、意见或建议,请登录顶顶电子网站:www.ddmcu.com,也可通过 E-mail: ddmcu@163.com 向我们提出,我们将为您提供超值的延伸服务,另外书中所需的实验板也可通过该网站购买。

最后要说明的是,本书中所有实例均采用了汇编语言进行编程,如果您喜欢用 C 语言,建议选用本书的姊妹篇 **《51 单片机 C 语言非常入门与视频演练》** 一书。

编著者

2010.3

目 录

第 1 章 51 单片机非常入门	1
1.1 单片机学前须知	1
1.1.1 常用数制介绍	1
1.1.2 数制的转换	2
1.1.3 数字电路基本概念	5
1.2 单片机概述	8
1.2.1 什么是单片机	8
1.2.2 单片机名称的由来	8
1.2.3 单片机与单片机系统	8
1.2.4 单片机的发展	9
1.2.5 单片机的分类	9
1.2.6 51 单片机家族简介	11
1.3 如何学习单片机	12
1.4 单片机的组成	14
1.4.1 单片机的内部结构	14
1.4.2 单片机的外部引脚	16
1.4.3 单片机存储器介绍	18
1.4.4 程序存储器	20
1.4.5 数据存储器	22
1.5 单片机的最小系统电路	27
1.5.1 单片机的工作电源	28
1.5.2 单片机的复位电路	28
1.5.3 单片机的时钟电路	28
第 2 章 51 单片机实验器材介绍与实验过程演示	29
2.1 DD-900mini 实验板介绍	29
2.1.1 DD-900mini 实验板硬件资源和接口	30
2.1.2 硬件电路介绍	30
2.1.3 仿真功能的使用	34
2.1.4 使用串口进行程序下载	35
2.1.5 笔记本电脑使用 DD-900mini 实验板	36
2.2 DD-51 编程器介绍与使用	37

2.2.1	DD-51 编程器介绍	37
2.2.2	DD-51 编程器使用	38
2.3	DD-F51 仿真器介绍	39
2.3.1	DD-F51 仿真器介绍	39
2.3.2	DD-F51 仿真器的组成与使用	39
2.4	51 单片机实验过程演示	40
2.4.1	编写程序	41
2.4.2	编译程序	45
2.4.3	仿真调试	46
2.4.4	烧写程序	49
2.4.5	脱机运行检查	49
第 3 章	51 单片机指令系统介绍与演练	50
3.1	指令系统概述	50
3.1.1	指令与指令系统	50
3.1.2	指令的格式	51
3.1.3	指令的字节数	51
3.1.4	指令的寻址方式	51
3.2	单片机指令系统介绍与演练	56
3.2.1	数据传送类指令介绍与演练	56
3.2.2	算术运算类指令介绍与演练	59
3.2.3	逻辑运算类指令介绍与演练	62
3.2.4	控制转移类指令介绍与演练	64
3.2.5	位操作类指令介绍与演练	67
3.2.6	伪指令介绍	69
3.3	应用广泛的延时程序与查表程序	70
3.3.1	定时(延时)程序	70
3.3.2	查表程序	73
第 4 章	单片机内部资源视频演练	75
4.1	中断系统介绍与视频演练	75
4.1.1	51 单片机的中断源	75
4.1.2	中断的控制	76
4.1.3	中断的响应	78
4.1.4	中断的撤除	79
4.1.5	视频演练 1——外中断练习	80
4.2	定时/计数器视频演练	81
4.2.1	什么是计数和定时	81
4.2.2	定时/计数器的组成	82
4.2.3	定时/计数器的寄存器	82

4.2.4	定时/计数器的工作方式	84
4.2.5	视频演练 2——定时实验	87
4.2.6	视频演练 3——计数实验	89
4.3	RS-232 串行通信视频演练	90
4.3.1	什么是 RS-232 串行通信	90
4.3.2	51 单片机串行口的结构	91
4.3.3	串行通信控制寄存器	92
4.3.4	串行口工作方式	93
4.3.5	视频演练 4——PC 通过 RS-232 和单片机通信	96
第 5 章	键盘接口电路视频演练	98
5.1	键盘接口电路基本知识	98
5.1.1	键盘的工作原理	98
5.1.2	键盘的扫描方式	99
5.1.3	DD-900mini 实验板中的独立按键	99
5.2	独立按键视频演练	99
5.2.1	视频演练 1——按键扫描方式练习	99
5.2.2	视频演练 2——可控流水灯	101
第 6 章	LED 数码管视频演练	104
6.1	LED 数码管介绍	104
6.1.1	LED 数码管的结构	104
6.1.2	LED 数码管的显示码	105
6.1.3	LED 数码管的显示方式	106
6.2	LED 数码管视频演练	108
6.2.1	视频演练 1——数码管动态扫描演示	108
6.2.2	视频演练 2——数码管电子钟	111
第 7 章	LCD 显示视频演练	118
7.1	字符型 LCD 介绍	118
7.1.1	字符型 LCD 引脚功能	118
7.1.2	字符型 LCD 内部结构	119
7.1.3	字符型 LCD 控制指令	121
7.1.4	字符型 LCD 与单片机的连接	124
7.2	字符型 LCD 视频演练	124
7.2.1	视频演练 1——1602 LCD 显示静止的字符串	124
7.2.2	视频演练 2——1602 LCD 显示从右向左移动的字符串	126
7.2.3	视频演练 3——1602 LCD 电子钟	128
第 8 章	时钟芯片 DS1302 视频演练	133
8.1	时钟芯片 DS1302 介绍	133

8.1.1	DS1302 概述	133
8.1.2	DS1302 的控制命令字	134
8.1.3	DS1302 的寄存器	134
8.1.4	DS1302 的数据传送方式	136
8.2	DS1302 数码管电子钟视频演练	136
8.2.1	实现功能	136
8.2.2	源程序	137
8.2.3	源程序解读	142
8.2.4	视频演示	142
第 9 章	单片机读/写 I ² C 总线视频演练	144
9.1	I ² C 总线介绍	144
9.1.1	I ² C 总线工作原理	144
9.1.2	I ² C 总线的电气结构	145
9.1.3	I ² C 总线器件的寻址方式	145
9.1.4	I ² C 总线数据的传输规则	145
9.1.5	I ² C 总线数据的读/写格式	146
9.1.6	I ² C 总线接口芯片 24C04 介绍	147
9.2	I ² C 总线接口芯片 24C04 视频演练	148
9.2.1	视频演练 1——具有记忆功能的计数器	148
9.2.2	视频演练 2——花样流水灯	152
第 10 章	温度传感器 DS18B20 视频演练	155
10.1	温度传感器 DS18B20 介绍	155
10.1.1	DS18B20 引脚功能	155
10.1.2	DS18B20 的内部结构	155
10.1.3	DS18B20 的指令	157
10.1.4	DS18B20 使用注意事项	158
10.2	DS18B20 LED 数字温度计视频演练	158
10.2.1	实现功能	158
10.2.2	源程序	158
10.2.3	源程序解读	163
10.2.4	视频演示	163
第 11 章	红外遥控视频演练	164
11.1	红外遥控基本知识	164
11.1.1	红外遥控系统	164
11.1.2	红外遥控的编码与解码	164
11.1.3	DD-900mini 实验板遥控电路介绍	165
11.2	红外遥控视频演练	166

11.2.1	视频演练 1——LED 数码管显示遥控器键值	166
11.2.2	视频演练 2——LCD 显示遥控器键值	170
第 12 章	单片机音乐发声视频演练	175
12.1	单片机音乐发声概述	175
12.1.1	音调的控制	175
12.1.2	音长的控制	176
12.2	单片机音乐发声视频演练	177
12.2.1	视频演练 1——单片机发出模拟枪声	177
12.2.2	视频演练 2——单片机发出模拟救护车声	178
12.2.3	视频演练 3——单片机发出模拟消防车声	180
12.2.4	视频演练 4——单片机唱歌	182
附录 A	51 单片机 111 条指令速查表	185
附录 B	ASCII 码表	188
参考文献		190

第1章 51 单片机非常入门

独具魅力的单片机能使您体会到电脑的真谛，您可以用单片机亲自动手设计智能玩具，可以设计不同的应用程序实现不同的功能。既有硬件制作又有软件设计，既动脑又动手。初级水平可开发智能玩具；中级水平可开发一些智能控制器，如电脑鼠、智能车、各种遥控模型；高级水平可开发工业控制单元、网络通信等，并用汇编语言或 C 语言设计应用程序。围绕单片机形成的电子产业的未来，将会为电子爱好者提供更广阔的天地。投身到单片机世界来，将使您一生受益。

1.1 单片机学前须知

在介绍单片机之前，我们有必要重温一下数制及数字电路中的有关概念。

1.1.1 常用数制介绍

数制就是记数法、进位制。目前，人们通用的数制是十进制，但使用十进制并非是天经地义的，它只不过是来源于远古时代用十指记数的一种约定俗成的习惯。事实上，在我们的生活中也有使用非十进制的实例，例如，日期、时间的表示和进位，以及英制度量衡等。下面就几种常用的数制分别进行介绍。

1. 十进制 (D)

数制所使用的数码的个数为基，数制的每一位所具有的值称为权。十进制是以 10 为基数的计数体制，即它所使用的代码为 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，共有 10 个。十进制的权是以 10 为底的幂，如 $10=1\times 10^1+0\times 10^0$ 。每个十进制数都可以用位权值表示，其中，个位的位权为 10^0 ，十位的位权为 10^1 ，百位的位权为 10^2 ，依此类推。

例如： $402.8=4\times 10^2+0\times 10^1+2\times 10^0+8\times 10^{-1}$

从数字电路的角度来看，采用十进制是不方便的。因为构成数字电路的基本思路是把电路的状态与数码对应起来，而十进制的 10 个数码必须有 10 个不同的而且能严格区分的电路状态与之对应起来，这样将在技术上带来许多困难，而且不经济，因此，在数字电路中一般不直接采用十进制，而采用二进制。

2. 二进制 (B)

二进制是用两个数码 0 和 1 表示的，而且是“逢二进一”，即 $1+1=10$ （读为“壹零”）。注意：这里的“10”与十进制数的“10”是完全不同的，它不代表“十”，右边的“0”表示 0 个 2^0 ，左边的“1”表示 1 个 2^1 ，即 $10=1\times 2^1+0\times 2^0$ ，可见，二进制的权是以 2 为底的幂。

使用二进制具有以下优点。

(1) 二进制只需要使用两种状态表示数字，容易实现

计算机是由电子元器件构成的，二进制在电气、电子元器件中最易实现。它只有两个数

字，用两种稳定的物理状态即可表达，而且稳定可靠。例如，磁化与未磁化，晶体管的截止与导通（表现为电平的高与低）等。而若采用十进制，则需要用 10 种稳定的物理状态分别表示 10 个数字，不易找到具有这种性能的元器件。即使有，其运算与控制的实现也极其复杂。

（2）二进制的运算规则简单

加法是最基本的运算，乘法是连加，减法是加法的逆运算（利用补码原理，还可以转化为加法运算，类似钟表拨针时的计算），除法是乘法的逆运算。其余任何复杂的数值计算也都可以分解为基本算术运算复合进行。为提高运算效率，在计算机中除采用加法器外，也直接使用乘法器。

我们知道，十进制的加法和乘法运算规则有上百条，用计算机的电路实现这么多运算规则是很复杂的。相比之下，二进制的算术运算规则非常简单，加法、乘法各仅 4 条，运算操作也比较方便。

（3）用二进制容易实现逻辑运算

数字电路不仅需要算术功能，还应具备逻辑运算功能，二进制的 0、1 分别可用来表示假（False）和真（True），用布尔代数的运算法则很容易实现逻辑运算。

但是，二进制也有一些缺点。例如，用二进制表示一个数时，位数多，使用起来不方便也不习惯。因此，在运算时，原始数据多用人们习惯的十进制，在送入计算机时，就必须将十进制数据转换成数字系统能接受的二进制数，而运算结束后再将二进制数转换为十进制数，表示最终结果。

3. 八进制（O）和十六进制（H）

由于使用二进制位数很多，不便于书写和记忆，因此，在数字计算机的资料中常采用八进制数或十六进制数来表示二进制数。上述十进制数和二进制数的表示法可以推广到八进制数和十六进制数。例如，八进制数采用 8 个数码，而且“逢八进一”。这种数制中有 8 个不同的数字：0, 1, 2, 3, 4, 5, 6, 7，它是以 8 为基数的计数体制。十六进制数采用 16 个数码，而且“逢十六进一”。这种数制中有 16 个不同的数字：0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A（对应于十进制数中的 10），B(11), C(12), D(13), E(14), F(15)，它是以 16 为基数的计数体制。

为便于记忆和理解，对二进制、八进制、十进制和十六进制进行了对照和比较，如表 1-1 所示。

表 1-1 二进制、八进制、十进制和十六进制对照比较情况

常用进制	英文表示符号	数码符号	进位规律	进位基数
二进制	B	0, 1	逢二进一	2
八进制	O	0, 1, 2, 3, 4, 5, 6, 7	逢八进一	8
十进制	D	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	逢十进一	10
十六进制	H	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	逢十六进一	16

1.1.2 数制的转换

1. 其他进制转换为十进制

其他进制转换为十进制的方法是将其他进制按权位展开，然后各项相加，得到相应的十

进制数。

【例 1】 将二进制数 $(11011.101)_2$ 转换成十进制数。

$$\begin{aligned}\text{解: } (11011.101)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125 \\ &= (27.625)_{10}\end{aligned}$$

【例 2】 将八进制数 $(136.524)_8$ 转换成十进制数。

$$\begin{aligned}\text{解: } (136.524)_8 &= 1 \times 8^2 + 3 \times 8^1 + 6 \times 8^0 + 5 \times 8^{-1} + 2 \times 8^{-2} + 4 \times 8^{-3} \\ &= 64 + 24 + 6 + 0.625 + 0.03125 + 0.0078125 \\ &= (94.6640625)_{10}\end{aligned}$$

【例 3】 将十六进制数 $(FF)_{16}$ 转换为十进制数。

$$\text{解: } (FF)_{16} = 15 \times 16^1 + 15 \times 16^0 = 240 + 15 = (255)_{10}$$

【例 4】 将十六进制数 $(13DF.B8)_{16}$ 转换成十进制数。

$$\begin{aligned}\text{解: } (13DF.B8)_{16} &= 1 \times 16^3 + 3 \times 16^2 + 13 \times 16^1 + 15 \times 16^0 + 11 \times 16^{-1} + 8 \times 16^{-2} \\ &= 4096 + 768 + 208 + 15 + 0.6875 + 0.03125 \\ &= (5087.71875)_{10}\end{aligned}$$

2. 将十进制转换成其他进制

十进制转换成其他进制时，需要分两部分进行，即整数部分和小数部分。

(1) 整数部分（基数除法）

把要转换的十进制数除以新的进制的基数，把余数作为新进制的最低位；

把上一次所得的商再除以新的进制基数，把余数作为新进制的次低位；

继续上一步，直到最后的商为零，这时的余数就是新进制的最高位。

(2) 小数部分（基数乘法）

把要转换数的小数部分乘以新进制的基数，把得到的整数部分作为新进制小数部分的最

高位；

把上一步所得的小数部分再乘以新进制的基数，把整数部分作为新进制小数部分的次高位；

继续上一步，直到小数部分变成零为止，或者达到预定的要求也可以。

【例 5】 把十进制数 $(25)_{10}$ 转换成二进制数。

解：由于二进制基数为 2，所以，逐次除以 2 取其余数（0 或 1），即

$$\begin{array}{rcl} 2 & \overline{) 25} & \\ 2 & \overline{) 12} & \cdots \cdots \cdots \text{余 } 1 \\ 2 & \overline{) 6} & \cdots \cdots \cdots \text{余 } 0 \\ 2 & \overline{) 3} & \cdots \cdots \cdots \text{余 } 0 \\ 2 & \overline{) 1} & \cdots \cdots \cdots \text{余 } 1 \\ 0 & & \cdots \cdots \cdots \text{余 } 1 \end{array}$$

所以， $(25)_{10} = (11001)_2$

【例 6】 把十进制数 $(173)_{10}$ 转换成二进制数。

解：由于二进制基数为 2，所以，逐次除以 2 取其余数（0 或 1），即

$$\begin{array}{r}
2 \overline{) 173} \\
2 \overline{) 86} \quad \cdots \cdots \text{余 } 1 \\
2 \overline{) 43} \quad \cdots \cdots \text{余 } 0 \\
2 \overline{) 21} \quad \cdots \cdots \text{余 } 1 \\
2 \overline{) 10} \quad \cdots \cdots \text{余 } 1 \\
2 \overline{) 5} \quad \cdots \cdots \text{余 } 0 \\
2 \overline{) 2} \quad \cdots \cdots \text{余 } 1 \\
2 \overline{) 1} \quad \cdots \cdots \text{余 } 0 \\
0 \quad \cdots \cdots \text{余 } 1
\end{array}$$

所以, $(173)_{10} = (10101101)_2$

【例 7】 将十进制数 $(25)_{10}$ 转换成八进制数。

解: 由于基数为 8, 逐次除以 8 取余数, 即

$$\begin{array}{r}
8 \overline{) 25} \\
8 \overline{) 3} \quad \cdots \cdots \text{余 } 1 \\
0 \quad \cdots \cdots \text{余 } 3
\end{array}$$

所以, $(25)_{10} = (31)_8$

【例 8】 将十进制 $(255)_{10}$ 转换为十六进制数。

解: 由于基数为 16, 逐次除以 16 取余数, 即

$$\begin{array}{r}
16 \overline{) 255} \\
16 \overline{) 15} \quad \cdots \cdots \text{余 } 15(\text{F}) \\
0 \quad \cdots \cdots \text{余 } 15(\text{F})
\end{array}$$

所以, $(255)_{10} = (\text{FF})_{16}$

【例 9】 将十进制小数 $(0.375)_{10}$ 转换成二进制数。

解:

$$\begin{array}{r}
0.375 \\
\times 2 \\
\hline
[0].750 \\
2 \\
\hline
[1].500 \\
2 \\
\hline
[1].000
\end{array}$$

所以, $(0.375)_{10} = (0.011)_2$

3. 二进制数与八进制数、十六进制数的相互转换

八进制数和十六进制数的基数分别为 $8=2^3$, $16=2^4$, 所以, 3 位二进制数恰好相当一位八进制数, 4 位二进制数相当 1 位十六进制数, 它们之间的相互转换是很方便的。

二进制数转换成八进制数的方法是从小数点开始, 分别向左、向右, 将二进制数按每 3 位一组分组 (不足 3 位的补 0), 然后写出每一组等值的八进制数。

【例 10】求 $(01101111010.1011)_2$ 的等值八进制数。

解: 二进制 001 101 111 010. 101 100

八进制 1 5 7 2 . 5 4

所以, $(01101111010.1011)_2=(1572.54)_8$

二进制数转换成十六进制数的方法和二进制数与八进制数的转换相似, 从小数点开始分别向左、向右将二进制数按每 4 位一组分组 (不足 4 位补 0), 然后写出每一组等值的十六进制数。

【例 11】将 $(1101101011.101)_2$ 转换为十六进制数。

解: 二进制 0011 0110 1011. 1010

十六进制 3 6 B. A

所以, $(1101101011.101)_2=(36B.A)_{16}$

八进制数、十六进制数转换为二进制数的方法可以采用与前面相反的步骤, 即只要按原来顺序将每一位八进制数 (或十六进制数) 用相应的 3 位 (或 4 位) 二进制数代替即可。

【例 12】将 $(678.A5)_{16}$ 转换为二进制数。

解: 十六进制 6 7 8 . A 5

二进制 0110 0111 1000 . 1010 0101

所以, $(678.A5)_{16}=(0110\ 0111\ 1000.1010\ 0101)_2$

1.1.3 数字电路基本概念

1. 数位与比特

码的位称为数位, 对于十进制码叫做十进制数位, 对于二进制码叫做二进制数位。二进制数位一般简称为 bit, 中文读做比特。例如, 某一个二进制码是 00100101, 该码共有 8 数位, 所以, 称为 8bit (比特)。

2. 字

用二进制数表示某一个数值或字符时, 该二进制数称为字, 英文是 Word。在数字系统电路中, 所有的信息, 包括数据、字母、符号、代表机器操作的指令或数据以及指令在存储器中的存放地址等, 都是以二进制代码表示的。作为一个整体来处理或运算的一组二进制数码, 称为一个字。字是二进制数的基本单位, 是数据总线宽度。

3. 字长

在微控制器中, 一个字的二进制位数称为字长。微控制器的字长有 1 位、4 位、8 位、16 位、32 位等。

4. 字节

字节是由一组二进制位形成的计算机的一种存储单位，用英文字母 **B** 表示，它可以表示一个字符，通常一个字节为 8 个二进制位。

5. 字内位的名称

字内各个位的名称是有规定的，具体规定如下：最高一位叫做 **MSB**，次高位叫做 **2SB**，第三位叫做 **3SB**，依此类推，最后一位叫做 **LSB**。

6. 二-十进制编码（BCD 码）

二-十进制编码也称 **BCD 码**，它是一个用 **4bit** 二进制码来表示十进制数的二进制码，即用 4 数位的二进制码来表示十进制数中的 0~9。如表 1-2 所示列出了几种常用 **BCD 码** 的编码方式。

表 1-2 几种常用的 BCD 码

十进制数	8421 码	5421 码	2421 码	余 3 码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	1000	1011	1000
6	0110	1001	1100	1001
7	0111	1010	1101	1010
8	1000	1011	1110	1011
9	1001	1100	1111	1100

这几种 **BCD 码** 中，**8421 BCD 码** 是最基本和最常用的 **BCD 码**，下面简要对 **8421 码** 进行说明。

从表 1-2 可以看出，十进制数中的 0~9 共 10 个数，每个数都用一个 **4bit** 的二进制码表示，而十进制数中的两位及两位以上的数，则采用每一位都用一个 **4bit** 的二进制码来表示。例如，123 用 **8421 BCD 码** 表示就是 0001 0010 0011，其中 0001 是表示百位数的 1，0010 表示的是十位数的 2，0011 表示的是个位数 3。

8421 BCD 码 的权自左至右为 8、4、2、1。具体地讲，8 是最高位（第四位）的权，4 是次高位（第三位）的权，2 是第二位的权，1 是最低位的权。

根据每一位的权，可以方便地计算出十进制数。例如，某一个二进制数码是 0111，则该数码就是十进制数中的 $0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 = 7$ 。

7. 什么是 ASCII 码

在数字系统中，除数字需要编码成为二进制码以外，各种字母和符号也必须用某种特定规则的二进制编码来表示。目前，国际上普遍采用的是 **ASCII 码**，其英文为 **American Standard Code for Information Interchange**，是美国标准信息交换码的简称。**ASCII 码** 用 7 位二进制数码来表示，故可表示 $2^7=128$ 种不同的字符，这其中包括了 26 个大小写英文字母、10 个十进制数字符号 0~9、7 个标点符号、9 个运算符号，以及 50 个其他符号等。

8. 什么是原码、反码和补码

(1) 原码

在生活中，表示数的时候一般都是把正数前面加一个“+”，负数前面加一个“-”，但是在数字设备中，机器是不认识这些的。一般规定，用二进制的最高位“0”表示“+”，用二进制的最高位“1”表示“-”，其余数位表示数的大小。

例如， $[+105]_{\text{原}}=01101001$ ， $[-105]_{\text{原}}=11101001$ $[+0]_{\text{原}}=00000000$ $[-0]_{\text{原}}=10000000$ 。

有符号的8位（一个字节）二进制数表示的范围为 $-127 \sim +127$ 。

(2) 反码

正数的反码与其原码相同，而负数的反码等于其绝对值各位求反。

例如， $[+105]_{\text{反}}=01101001$ ， $[-105]_{\text{反}}=10010110$ ， $[+0]_{\text{反}}=00000000$ ， $[-0]_{\text{反}}=11111111$ 。

(3) 补码

在介绍补码概念之前，有必要先介绍补数的概念。以天天见到的时钟为例，设当前的准确时间是下午2点，可时钟却停在10点的位置上，将时钟修正到下午2点位置有两种方法：一是将时钟顺时针方向拨4h；二是将时钟逆时针方向拨8h，其结果是一样的。上述两种拨钟方法可用下式表示：

$$10+4=14 \text{ 点}=2 \text{ 点；}$$

$$10-8=2 \text{ 点。}$$

14点就是下午2点，这是因为时钟按12h为一个周期计数，12称为模，即时钟系统的模是12，14和2是以12为模的同余数。在12为模的系统中，4和8互为补数，此外，3和9，2和10等都是互为补数，它们的共同特点是两数相加等于12。因此，要求一个数的补数，用模减去这个数就可以了。

有了上述补数的概念之后，就容易理解补码的概念了，在数字系统电路中的补数就是补码。

一般来说，正数的补码与其原码相同；负数的补码是把其原码除符号位外的各位先求其反码，然后在最低位加1。

例如， $[+105]_{\text{补}}=01101001$ $[-105]_{\text{补}}=10010111$ $[0]_{\text{补}}=00000000$ $[-1]_{\text{补}}=11111111$ 。

9. 数字信号的传输速率

数字信号的传输速率也称为波特率或比特率，是指每秒所传输数据的位数，它的单位是位/秒（bit/s）。例如，某通信系统每秒传输120个字符，而每个字符格式包含10个代码位（1个起始位、1个停止位、8个数据位），这时，传输的波特率为

$$10 \times 120 \text{ bit/s} = 1200 \text{ bit/s}$$

第一位代码传输的时间 t_d 为波特率的倒数，即

$$t_d = \frac{1}{1200} = 0.883 \text{ ms}$$

波特率是衡量传输通道频宽的指标，它和传输数据的速率并不一致，如本例中，因为除掉起始位和停止位，每1位数据实际上只有8位，所以，数据的传输速率为

$$8 \times 120 \text{ bit/s} = 960 \text{ bit/s}$$