



高职高专通用教材

微机原理与接口技术

王 丰 王兴宝 编著



北京航空航天大学出版社

高职高专通用教材

微机原理与 接口技术

王 丰 王兴宝 编著



北京航空航天大学出版社

内 容 简 介

本书较系统地介绍了 16 位微型计算机的基本原理及接口技术,并对计算机的最新发展做了适度介绍。全书共分 15 章,主要内容包括:基础知识、微型计算机的组成及工作原理、指令系统、MASM 伪指令系统、汇编语言程序设计、8086 微处理器结构、半导体存储器、输入/输出技术、中断、并行接口芯片 8255A、串行通信、DMA 控制器 8237A、其他常用接口、PC 机系统结构及总线技术和实验指南等。

本书内容的组织力求深入浅出,以培养学生应用能力为主要目标,以基本知识和基本技术介绍为主,内容丰富,实用性强,可作为高职高专院校计算机专业的教材,以及自动化、机电等非计算机专业的本科生教材,也可供工程技术人员、自学人员参考使用。

图书在版编目(CIP)数据

微机原理与接口技术/王丰等编著. —北京:北京航空航天大学出版社,2005. 3
ISBN 7-81077-593-6

I. 微… II. 王… III. ①微型计算机—理论②微型计算机—接口 IV. TP36

中国版本图书馆 CIP 数据核字(2005)第 008099 号

微机原理与接口技术

王 丰 王兴宝 编著
责任编辑:张冀青

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpess@263.net

北京松源印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:24.5 字数:549 千字

2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷 印数:5 000 册

ISBN 7-81077-593-6 定价:33.00 元

前言

微机原理与接口技术是计算机及相关专业的一门重要专业基础课,是计算机组成原理、计算机系统结构和计算机控制系统等课程的前导课。

本书以 PC/XT 为主要背景机,以 8086/8088 CPU 为基础来介绍汇编语言程序设计、微机工作原理及接口开发的相关知识。虽然 Pentium 机和 Windows 操作系统是当前 PC 机的主流,但不论是 Pentium 机还是 Windows,其复杂性使其不适合应用于初学者的教学。对于初学者来说,首先要掌握的是计算机基本工作原理和框架结构,而不是过多过杂的技术细节。同时,考虑到计算机本身的发展,本书中对一些新技术也做了相应的介绍。

全书内容根据教育部高职高专规划教材指导思想与原则的要求,充分考虑了专科学生的培养目标和教学特点。在内容的组织上,本着由浅入深、循序渐进的原则,注重基本知识和基本概念的介绍,结合实例重点介绍实用性较强的内容,对难度过大的内容只做少量介绍,使学生有的放矢,利于掌握所学内容。

考虑到非计算机类专业的学时限制,即不能提前一学期开出本课程的前导课《数字电路》,而是常出现《数字电路》与《微机原理》同时开课的情况,所以本书的第 1 章对本课程所需的前导知识做了较系统的总结,以满足这种情况下的教学需求。这也使得本书内容自成体系,便于自学者使用。另外,依据作者十多年的教学经验,汇编语言的学习是本课程的主要难点。为此,本书的第 3 章对主要指令做了尽可能详细的讲解,另外有意识地将指令分散于多章中讲解,以求分散难点。最后,专门用第 15 章对上机内容进行了具体的示范讲解,以使初学者能够领会并掌握汇编语言。

全书共分 15 章,内容包括:基础知识、微型计算机的组成及工作原理、指令系统、MASM 伪指令系统、汇编语言程序设计、8086 微处理器结构、半导体存储器、输入/输出技术、中断、并行接口芯片 8255A、串行通信、DMA 控制器 8237A、其他常用接口、PC 机系统结构及总线技术和实验指南等内容。书中有少量内容(如:浮点数的格式及表示范围等)对高职高专学生可能略有难度,但考虑到计算机等级考试,专升本考试等的需要,因此加了进来,教师在授课时可适当节选。

书中凡以完整格式给出的程序均可直接使用,但须注意在向计算机内录入程

序时,所有标点符号均应为西文符号;而书中为了显示清晰,故将程序代码中的逗号、引号和分号等改用了中文符号。

王丰编写了本书的第2~15章,王兴宝编写了本书的第1章及附录,全书由王丰统稿。此外,参加本书编写及资料整理的还有孙连军、高政、孙凤美、王海鹏、陈光军、谭树芹等同志,本书由北方工业大学田红芳副教授担任主审。

本书编写过程中,一直得到了北京航空航天大学出版社教材编辑室蔡喆主任的大力支持和指导,在此一并表示衷心感谢。

由于作者水平有限,书中错误和不足在所难免,恳请广大读者批评指正。

王 丰
2005年1月

目 录

第 1 章 基础知识

1.1 数 制	1
1.1.1 二进制	1
1.1.2 十六进制	4
1.1.3 八进制	7
1.2 码 制	7
1.2.1 带符号数的编码	7
1.2.2 数的定点表示与浮点表示	11
1.2.3 ASCII 码与 BCD 码	13
1.2.4 如何理解计算机中数的多种表示形式	14
1.3 常用数字逻辑器件	15
1.3.1 基本门电路	15
1.3.2 触发器	18
1.3.3 译码器	21
习题一	22

第 2 章 微型计算机的组成及工作原理

2.1 微型计算机系统及其工作原理	23
2.1.1 概 述	23
2.1.2 硬件系统结构	24
2.1.3 软件系统结构	27
2.1.4 微型计算机的工作原理	31
2.2 PC 机的编程结构	32
2.2.1 8086 CPU 的内部结构	32
2.2.2 8086/8088 的寄存器结构	34
2.2.3 内存储器基础	38
2.2.4 内存分段	39
习题二	42

第3章 指令系统

3.1 寻址方式	43
3.1.1 指令格式及操作数类型	43
3.1.2 8086/8088 的寻址方式	44
3.2 8086/8088 指令系统	48
3.2.1 数据传送指令	48
3.2.2 算术运算指令	56
3.2.3 逻辑运算及移位指令	62
3.2.4 转移控制指令	70
3.2.5 处理器控制指令	75
3.2.6 简单程序设计举例	76
习题三	80

第4章 MASM 伪指令系统

4.1 概 述	83
4.2 汇编语言语句种类与格式	84
4.2.1 汇编语言语句的种类	84
4.2.2 汇编语言语句的格式	84
4.3 常用伪指令	88
4.3.1 符号定义伪指令	88
4.3.2 数据定义伪指令与变量	88
4.3.3 标号及其属性	92
4.3.4 析值/属性操作符及符号名定义语句	93
4.3.5 段定义伪指令	97
4.4 程序的段结构	99
4.5 地址表达式	103
4.6 宏定义与宏调用	104
4.6.1 宏定义	104
4.6.2 宏调用	105
4.6.3 实 例	105
习题四	106

第 5 章 汇编语言程序设计

5.1 汇编语言软件开发步骤及输入/输出问题	108
5.1.1 汇编语言软件开发步骤	108
5.1.2 汇编语言编程时的输入/输出问题	108
5.2 顺序程序设计	111
5.2.1 字符串的输入/输出	111
5.2.2 BCD 码的显示	112
5.2.3 查表	113
5.3 分支程序设计	114
5.3.1 基本分支程序设计	114
5.3.2 跳转表法实现分支	115
5.4 循环程序设计	116
5.4.1 循环程序的结构	116
5.4.2 循环程序设计	117
5.5 专题应用程序设计	119
5.5.1 子程序设计	119
5.5.2 代码转换原理及程序设计	127
5.5.3 BCD 码运算及其程序设计	133
5.5.4 串操作指令及其程序设计	135
习题五	140

第 6 章 8086 微处理器结构

6.1 8086 微处理器工作模式与引脚功能	141
6.1.1 8086 的工作模式	141
6.1.2 两种工作模式下共用引脚	141
6.1.3 最小模式引脚	143
6.1.4 最大模式引脚	144
6.1.5 8088 与 8086 的差别	145
6.2 系统组成	146
6.2.1 锁存与收发	146
6.2.2 最小模式下的系统组成	148
6.2.3 最大模式下的系统组成	148
6.3 8086 的总线操作与时序	149

6.3.1 指令周期、总线周期与 T 状态	149
6.3.2 8086 总线读写周期	150
6.4 从 80286 到 Pentium 微处理器简介	154
习题六	156

第 7 章 半导体存储器

7.1 概 述	157
7.1.1 半导体存储器的分类与技术指标	157
7.1.2 静态读写存储器 SRAM	159
7.1.3 动态读写存储器 DRAM	163
7.2 CPU 与存储器的连接	165
7.2.1 主存容量的扩展方法	165
7.2.2 存储器的译码	168
7.2.3 8088 与存储器连接实例	171
7.2.4 8086 与存储器的连接	171
7.3 只读存储器及新型存储器	175
7.3.1 EPROM 的使用	175
7.3.2 E ² PROM 使用及编程	176
7.3.3 新型存储器简介	178
习题七	181

第 8 章 输入/输出技术

8.1 有关接口的基本概念	182
8.1.1 接口的概念与功能	182
8.1.2 接口信号的分类及基本结构	183
8.1.3 端口地址及编址方式	184
8.1.4 8086/8088 的输入/输出指令	185
8.1.5 端口地址的译码问题	186
8.2 接口数据的传送方式	187
8.2.1 无条件传送方式	187
8.2.2 查询传送方式	187
8.2.3 中断传送方式	190
8.2.4 DMA 传送方式	190
习题八	192

第 9 章 中 断

9.1 中断概述	193
9.1.1 中断的概念	193
9.1.2 中断源	193
9.1.3 中断类型	194
9.1.4 中断类型号	195
9.1.5 中断矢量表	195
9.2 中断处理过程	196
9.2.1 中断请求	196
9.2.2 中断响应	199
9.2.3 中断处理	200
9.2.4 中断返回	200
9.3 可编程中断控制器 8259A	201
9.3.1 8259A 的结构与引脚	202
9.3.2 8259A 中断响应时序	205
9.3.3 8259A 的工作方式	206
9.3.4 8259A 的命令字	208
9.3.5 8259A 的编程及其在 PC 机中的应用	214
9.4 中断服务程序的编写	215
9.4.1 中断服务程序编写的基本问题	215
9.4.2 中断服务程序实例	217
习题九	220

第 10 章 并行接口芯片 8255A

10.1 8255A 概述	222
10.1.1 8255A 的主要特性	222
10.1.2 8255A 的内部结构	222
10.1.3 8255A 的引脚	223
10.2 8255A 的控制字	224
10.2.1 工作方式控制字	224
10.2.2 C 口按位置位/复位控制字	225
10.3 8255A 的工作方式	226
10.3.1 方式 0: 基本输入/输出	226

10.3.2 方式 1:带选通的输入/输出	227
10.3.3 方式 2:带选通的双向输入/输出	229
10.4 8255A 的编程	230
习题十	235

第 11 章 串行通信

11.1 串行通信概述	236
11.1.1 串行通信的特点	236
11.1.2 串行通信的基本概念	236
11.1.3 串行接口概述	240
11.2 RS-232C 串行接口标准	242
11.2.1 电气特性	242
11.2.2 接口信号功能	243
11.2.3 信号线的连接	245
11.3 异步通信接口芯片 8250	246
11.3.1 INS 8250 的结构和外部特性	246
11.3.2 INS 8250 的寄存器	248
11.3.3 INS 8250 的编程	255
习题十一	261

第 12 章 DMA 控制器 8237A

12.1 DMA 控制器概述	262
12.1.1 DMAC 的结构与作用	262
12.1.2 DMAC 内部的寄存器	263
12.2 Intel 8237A 概述	264
12.2.1 8237A 的特点	264
12.2.2 8237A 的引脚	264
12.2.3 8237A 的内部结构	266
12.2.4 8237A 的寄存器	267
12.3 8237A 的软件命令	273
12.4 8237A 的工作时序	273
12.4.1 正常 DMA 读写操作	274
12.4.2 扩展写与压缩时序	275
12.5 8237A 的初始化及其在 PC 机中的应用	276

12.5.1 8237A 的初始化	276
12.5.2 8237A 在 PC 机中的应用	277
习题十二	278

第 13 章 其他常用接口

13.1 定时/计数器 8253	279
13.1.1 8253 定时/计数器概述	279
13.1.2 8253 的工作方式	281
13.1.3 8253 的控制字与编程	284
13.2 A/D 与 D/A 接口	287
13.2.1 概 述	287
13.2.2 D/A 转换接口	288
13.2.3 A/D 转换接口	294
13.3 人机交互设备	301
13.3.1 键 盘	302
13.3.2 LED 显示器	304
13.3.3 打印机接口	306
13.3.4 CRT 显示器及其接口	309
习题十三	315

第 14 章 PC 机系统结构及总线技术

14.1 PC 机的软件系统	316
14.1.1 DOS 的功能、结构及引导过程	316
14.1.2 DOS 功能调用	317
14.1.3 磁盘文件管理	318
14.1.4 BIOS 中断功能调用	324
14.2 PC/XT 硬件结构	329
14.2.1 PC/XT 主板结构	329
14.2.2 CPU 子系统	329
14.2.3 接口部件子系统	330
14.2.4 ROM 子系统和 RAM 子系统	333
14.3 总线技术	334
14.3.1 总线概述	334
14.3.2 典型总线简介	336

14.3.3 PC/XT 总线	340
习题十四	343

第 15 章 实验指南

15.1 《微机原理与接口技术》课程上机简介	344
15.1.1 汇编语言部分的上机	344
15.1.2 硬件部分的上机	346
15.2 汇编语言上机指导	348
15.2.1 DEBUG 使用详解	348
15.2.2 MASM 使用详解	360
15.2.3 LINK 使用详解	361
15.2.4 用 DEBUG 调试程序示例	363
15.2.5 程序员工作平台 PWB 介绍	369

附 录

附录 A ASCII 码表	372
附录 B 中断向量一览表	373
附录 C 8086/8088 指令速查表	374
附录 D MASM 出错信息表	375

参考文献

第 1 章 基础知识

本章内容是学习计算机原理的基础,介绍数据在计算机内的表示方法,计算机接口开发中常用到的数字逻辑器件以及一个最简单的计算机是如何工作的。1.1 和 1.2 小节是对《数字电路》中相关内容的简单复习和回顾。1.3 小节内容是为详细学习 PC 机的工作原理奠定基础。

1.1 数 制

数制研究日常生活中的数据在计算机内是如何表示的,这是使用计算机进行数据运算的前提。日常生活中最常用的是十进制数,但这样的数在计算机内表示起来比较困难,须将其转换为等价的二进制数形式。原因是二进制数每位的状态少(只有 0、1 两种,十进制数每一位都有 0~9 十种可能的状态),状态越少,电路越简单而且越可靠。

例如:若用某电子器件上的电压表示一位数字,电源电压为 5 V。对十进制数来说,需将 5 V 的电压划分为十个等份,依次表示 0~9 这十种状态,用 0 V 表示数字 0,0.5 V 表示数字 1,1 V 表示数字 2……这样不但准备十种电压值比较困难,而且抗干扰性差。如果由于干扰,器件上的电压值由 0.5 V 变为 0.75 V,则计算机无法确定该器件表示的是数字 1 还是数字 2。对于二进制数,则无此弊端,只需用 0 V 表示数字 0,用 5 V 表示数字 1,大大简化了电路,而且,干扰电压也不可能使 0 V 和 5 V 难以辨认,因而提高了可靠性。

1.1.1 二进制

1. 十进制数的特点及基数、权的概念

为了理解二进制数的特点,先看一下十进制数的特点:

① 有 10 个独立的符号 0~9,这些符号叫作数码,数码的个数叫作基数,十进制数的基数是 10。

② 逢十进一。

③ 任意一个十进制数 $a_n a_{n-1} \cdots a_1 a_0 . b_1 b_2 \cdots b_m$ 的值是:

$$a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10^1 + a_0 \times 10^0 + b_1 \times 10^{-1} + b_2 \times 10^{-2} + \cdots + b_m \times 10^{-m}$$
式中, n 是整数位数, m 是小数位数。 a_i, b_j 为 0~9 中的一个。

在一个数中,每个数码表示的值不仅取决于数码本身,还取决于它所处的位置,即个位、十位、百位……每一位有各自的权。例如: $158 = 1 \times 10^2 + 5 \times 10^1 + 8 \times 10^0$ 。其中 $10^2, 10^1, 10^0$ 分

别为百位、十位和个位的权。

2. 二进制数的特点

对比十进制数,二进制数有以下特点:

① 有两个独立的符号 0 和 1。二进制数的基数是 2。

② 逢二进一。

③ 任意一个二进制数 $a_n a_{n-1} \cdots a_1 a_0 . b_1 b_2 \cdots b_m$ 的值是:

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_m \times 2^{-m}$$

式中, n 是整数位数, m 是小数位数。 a_i, b_j 为 0, 1 中的一个。

例如: $101.01\text{B} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 5.25$

为了不与十进制数混淆,二进制数在其尾部加字母 B(Binary) 或者加下标 2 表示。例如: 10100101B 、 10100101_2 或 $(10100101)_2$ 。

相应地,十进制数可在尾部加字母 D(Decimal) 或下标 10。例如: 12345D 、 12345_{10} 或 $(12345)_{10}$ 。另外,十进制是默认的数制,当一个数后面没有表示其进制的标记时,该数当作十进制数处理。

3. 十进制数转换为二进制数

十进制数转换为二进制数时,整数部分与小数部分的转换方法不同,对于这两部分应分别转换,然后将得到的两部分二进制数合并起来就得到转换后的结果。

(1) 整数部分的转换

方法:基数除法。

规则:除 2 取余,逆序排列。将十进制整数不断除以 2,同时记下每次所得余数,直到商为 0 为止。最后,将余数按相反的顺序排列即可(最先的余数在最低位,最后的余数在最高位)。

例 1.1 将十进制数 20 转换为二进制数。如图 1-1 所示。

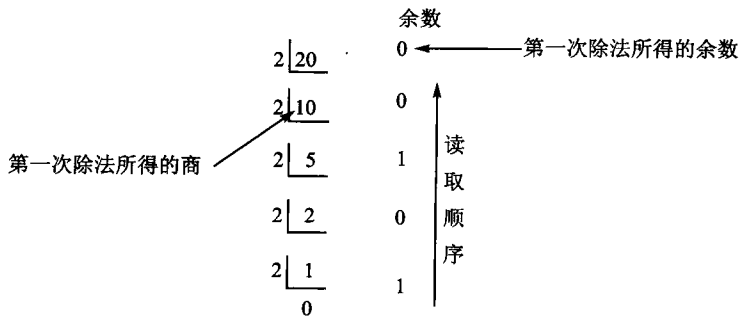


图 1-1 十进制整数转换为二进制数

$$20 = 10100\text{B}$$

(2) 小数部分的转换

方法:基数乘法。

规则:乘2取整,顺序排列。将十进制小数不断乘以2,同时记下每次所得的整数,直到小数部分为0或达到要求精度为止。最后,将整数按记下的先后顺序排列即可(最先的整数在最高位,最后的整数在最低位)。

例 1.2 将十进制数 0.625 转换为二进制数。如图 1-2 所示。

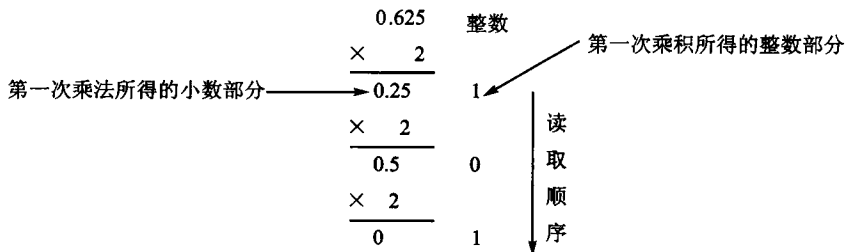


图 1-2 十进制小数转换为二进制数

$$0.625 = 0.101\text{B}$$

注意:一个二进制数可以准确地转换为十进制数,而一个带小数的十进制数不一定能够准确地用二进制数来表示。例如:十进制数 1.3 的小数部分就无法精确地转换为二进制数,只能取前面的一部分数值进行近似表示。

例 1.3 将十进制数 20.625 转换为二进制数。

将整数部分 20 和小数部分 0.625 分别进行转换,见例 1.1 和例 1.2,最后将两部分合并起来即可。

$$20.625 = 10100.101\text{B}$$

4. 二进制数转换为十进制数

转换方法是依据二进制数的第三个特点。

二进制数 $a_n a_{n-1} \cdots a_1 a_0 . b_1 b_2 \cdots b_m$ 的值是:

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_1 \times 2^1 + a_0 \times 2^0 + b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots + b_m \times 2^{-m}$$

例 1.4 将二进制数 1101.101 转换为十进制数。

$$1101.101\text{B} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 8 + 4 + 1 + 0.5 + 0.125 = 13.625$$

5. 二进制数运算

二进制数的运算规则同十进制数一样,只是逢二进一。如图 1-3 所示。

注意:当运算由计算机进行时,对于加减法,两数的位数必须相等,并且和或差的位数也与之相等。若加法的最高位向前有进位,则进位自动丢弃;而减法运算时若不够减,则最高位

$$\begin{array}{r}
 1101 \\
 + 1011 \\
 \hline
 11000
 \end{array}
 \qquad
 \begin{array}{r}
 1101 \\
 - 1011 \\
 \hline
 0010
 \end{array}
 \qquad
 \begin{array}{r}
 1101 \\
 \times 10 \\
 \hline
 0000 \\
 1101 \\
 \hline
 11010
 \end{array}
 \qquad
 \begin{array}{r}
 1101 \\
 10 \overline{) 11010} \\
 \underline{10} \\
 10 \\
 \underline{10} \\
 010 \\
 \underline{10} \\
 0
 \end{array}$$

图 1-3 二进制数运算

自动从前面取得一个借位。如: $1000 + 1000$ 应得结果为 10000, 但实得结果为 0000。0000—1000 实际进行的运算是 $10000 - 1000$, 因而得 1000。对于这个问题的纠正, 在 2.2.2 小节中标志寄存器部分进行了说明。

6. 相关概念及术语

位(bit, 又称比特): 指二进制数字的 1 位。

字节(Byte): 8 位二进制数为 1 个字节, 用 1 B 表示。

字(Word): 16 位二进制数为 1 个字, 即 1 个字 = 2 B。

注意:“字”与计算机的“字长”不是一回事。字是衡量二进制数长度(位数)的单位, 1 个字固定为 16 位二进制数的长度。字长指的是计算机的 CPU 一次能进行几位二进制数的运算, 是衡量计算机性能的一个指标。对不同的 CPU 而言, 字长也不一样。如: 8086 字长为 16 位, 286 字长为 16 位, 386、486 字长为 32 位。

除了以上三个基本单位外, 在描述存储器的存储容量时还常用如下几个单位:

$1 \text{ KB} = 2^{10} \text{ B} = 1\,024 \text{ B}$ (注意: 计算机中“K”代表 1 024, 而不是 1 000)

$1 \text{ MB} = 2^{20} \text{ B} = 2^{10} \text{ KB}$

$1 \text{ GB} = 2^{30} \text{ B} = 2^{10} \text{ MB}$

$1 \text{ TB} = 2^{40} \text{ B} = 2^{10} \text{ GB}$

除以上几个术语外, 还常遇到的一个问题是: n 位二进制数能表示多少个数?

通常, 1 位十进制数能表示 10 个数(0~9), 2 位能表示 100 个数(0~99)。由于每位数字可能有 10 种状态, 故 n 位十进制数能表示 10^n 个数; 同理, 二进制数中每位数字只能有两种状态, 因而 n 位二进制数可表示 2^n 个数(0~ $2^n - 1$)。

1.1.2 十六进制

十进制数转换为二进制数后, 往往要变长许多倍。例如, 数“8”在十进制下是 1 位, 而在二进制下是 4 位(1000B), 编程时这给书写带来了极大的不便, 为了解决这一问题, 人们引入了