

Microcomputer System and Interface Technology

仇玉章 主编

孙力娟 编写
高翔

微型计算机系统

接口技术

江苏
科学技术
出版社



微型计算机系统与接口技术

主 编 仇 玉 章
编 写 孙 力 娟
高 翔

江苏科学技术出版社

内 容 提 要

本书以 Intel 8088 微处理器为背景,讲述了汇编语言程序设计、微机系统和接口技术。

全书共分十四章,内容主要包括:8088 指令系统以及 286 以上高档微机的加强指令、汇编语言程序设计、常用总线标准、输入/输出系统、中断系统、串行通信、并行通信、DMA 传送、数/模和模/数转换、汇编语言高级编程。

本书注重实用,在有关接口的各章节中,均配有应用实例。

本书可作为高等院校计算机专业汇编语言程序设计、微机原理及应用、微机系统及接口技术等方面的教材,也可供从事微机应用的工程技术人员参考。

微型计算机系统和接口技术

编 著 仇玉章
责任编辑 梁海哨

出版发行 江苏科学技术出版社
 (南京市中央路 165 号,邮编:210009)

经 销 江苏省新华书店
印 刷 赣榆县印刷厂

开 本 787×1092 毫米 1/16
印 张 25
字 数 620,000
版 次 1997 年 5 月第 1 版
印 次 1997 年 5 月第 1 次印刷
印 数 1—5,000 册

标准书号 ISBN 7—5345—2324—9/TP·50
定 价 28.00 元

我社图书如有印装质量问题,可随时向承印厂调换。

前 言

由于计算机应用的不断普及,掌握微机系统知识及接口技术就显得十分重要。

本书编写的指导思想是:立足系统,面向应用,用汇编语言程序设计贯穿各个章节。本书不论是在汇编语言程序设计还是在接口技术的应用方面,内容力求全面,有一定的深度,并具有较强的实用性,是编者多年教学和科研的经验总结。为了深入理解微机系统和接口技术,本书还编写了较多的软件和硬件实验课题。

全书包括三部分:汇编语言程序设计,微机系统和接口技术。

第一、二章介绍了微机系统的基础知识。

第三章深入讲解了 8088 的指令系统,以及 286 以上高档微机的加强指令。

第四章比较全面地介绍了宏汇编语言的若干语法规则。

第五章讲述了汇编语言程序设计的设计方法。

第六章简要介绍了 16 位微机系统的组成原理、典型时序以及微机系统的常用总线标准。

第七章概述了微机系统的输入/输出系统,并重点介绍了 8253 定时/计数器以及在微机系统中的应用。

第八章从硬件和软件两个方面详细介绍了 PC/XT、PC/AT 机的中断系统及其应用。

第九章详细介绍了串行异步通信和接口芯片 8250、8251,以及在系统环境下进行串行通信的程序设计方法。

第十章着重讲述并行接口芯片 8255 及其应用。

第十一章重点讲述 8237DMA 控制器,以及实现 DMA 传送的方法。

第十二章简要介绍数/模、模/数转换的原理以及 DAC0832、ADC0809 的应用。

第十三章为汇编语言高级编程,是汇编语言程序设计的提高性章节。本章系统地讲述了驻留程序的设计方法及解除驻留的手段,讲述了菜单程序的设计方法。“分屏幕显示全双工通信”介绍了串行通信的实用性编程方法,本章还讲述了程控交换机的收号原理以及编程方法。

在第十四章中,为了深入理解微机系统和接口技术的各个主要环节,我们编写了若干个软件和硬件的实验课题,这些课题涉及面广,有一定难度。

本书第一、二、三章由高翔同志编写,第四、五、八、十二、十三、十四章,以及 § 3.6, § 7.5, § 9.4, § 10.4, § 11.4 由仇玉章同志编写,第六、七、九、十、十一章由孙力娟同志编写,全书由仇玉章同志主编。

谢秋丽同志绘制全书的图表,周宁宁同志、吴晓芬同志和本系的部分同学协助完成了大部分书稿的录入工作,编者对他们深表谢意。

由于编者水平所限,书中难免有错漏之处,敬请读者批评指正。

目 录

第一章 计算机中的数制和码制	1
§ 1.1 数制	1
一、常用计数制	1
二、数制换算	2
三、字符编码	3
§ 1.2 码制	4
一、机器数的原码、补码和反码	5
二、整数补码的运算	5
三、无符号数	7
§ 1.3 浮点数的基本概念	7
一、浮点数	7
二、浮点机器数	8
三、浮点数的数值范围	9
习题	10
第二章 计算机基础	11
§ 2.1 计算机系统组成	11
一、计算机的硬件组成	11
二、计算机系统的软件	12
§ 2.2 微型计算机	12
一、微型机的硬件结构	12
二、微型机的基本配置	13
§ 2.3 8088 CPU	14
一、8088 的寄存器结构	14
二、8088 的功能结构	15
三、8088 的存储器组织	15
第三章 8088 指令系统	18
§ 3.1 概述	18
§ 3.2 8088 寻址方式	19
一、立即寻址	19
二、寄存器寻址	19
三、直接寻址	20
四、寄存器间接寻址	20

五、变址寻址	21
六、基址+变址寻址	21
§ 3.3 寻址方式的段约定和段超越	21
一、段约定	21
二、段超越	22
§ 3.4 8088 标志寄存器	22
§ 3.5 8088 指令系统	23
一、传送类指令	24
二、算术运算指令	28
三、转移和调用指令	35
四、逻辑运算指令	41
五、串操作指令	44
六、处理机控制指令	50
§ 3.6 80286 加强指令	51
习题	54
 第四章 宏汇编语言	 55
§ 4.1 汇编源程序的语句类型	55
§ 4.2 宏汇编的基本语法	56
一、标号和变量	56
二、常数	57
三、数值运算符	58
四、方括号运算符和地址表达式	58
五、修改属性的运算符(PTR)	59
六、返回属性或数值的运算符	59
§ 4.3 宏汇编的常用伪指令	60
一、符号定义伪指令	60
二、字节定义伪指令	61
三、字定义伪指令	61
四、双字定义伪指令	62
§ 4.4 宏汇编基本语句	62
一、段定义语句	62
二、段约定语句	64
三、过程定义语句	65
四、定位语句	65
五、公用符号说明语句	66
六、外部符号说明语句	66
七、汇编结束语句	66
§ 4.5 源程序的编程格式	69
一、EXE 文件的汇编格式	69

二、COM 文件的汇编格式	70
三、EXE 文件和 COM 文件的内存映像	71
四、程序段前缀	72
五、返回 DOS 的其它方法	73
§ 4.6 汇编语言程序的上机过程	76
一、源程序的编辑、编译和链接	76
二、MASM 和 LINK 的使用方法	77
三、COM 文件的生成	78
四、批处理文件的使用	79
§ 4.7 宏指令的定义与调用	79
一、无参数的宏定义伪指令	79
二、宏调用	80
三、带有哑参数的宏定义伪指令	80
四、带有哑参数的宏指令调用	81
五、LOCAL 伪指令	81
六、宏定义嵌套	84
七、宏指令和子程序的区别	85
§ 4.8 条件汇编	85
一、条件汇编的基本格式	85
二、常用的条件汇编语句	85
三、应用举例	85
第五章 汇编语言程序设计	87
§ 5.1 DOS 系统 I/O 功能调用	87
§ 5.2 BIOS 键盘输入功能调用	91
§ 5.3 CRT 显示器	92
§ 5.4 BIOS 屏幕显示功能调用	94
§ 5.5 分支程序	96
一、简单分支	97
二、复合分支	97
三、多分支	99
§ 5.6 循环程序	100
§ 5.7 子程序	101
§ 5.8 代码转换	106
§ 5.9 数值计算	116
§ 5.10 数据处理	120
§ 5.11 延时程序	135
一、最简单的延时程序	135
二、使用 DOS 系统功能调用设计延时程序	136
三、使用 BIOS 功能调用设计延时程序	137

§ 5.12 字符串的动画显示	137
§ 5.13 模块化程序设计	140
一、PUBLIC 伪指令	141
二、EXTRN 伪指令	141
三、模块化程序设计举例	141
四、模块化设计应注意的问题	147
§ 5.14 磁盘文件 I/O	148
一、DOS 磁盘文件操作功能调用	149
二、磁盘文件操作程序设计	151
习题	156
第六章 总线和时序	157
§ 6.1 8088 的引脚功能	157
一、最小/最大模式公用引脚	158
二、8088 最小模式系统	159
三、最大模式系统	160
§ 6.2 8088 的典型时序	162
一、指令周期、总线周期和 T 状态	162
二、存储器读周期	162
三、存储器写周期	164
§ 6.3 总线标准	164
一、PC 总线	165
二、AT 总线	166
三、EISA 总线	169
四、局部总线	169
第七章 输入/输出系统	170
§ 7.1 输入/输出接口概述	170
一、CPU 与输入/输出设备交换信息的途径	170
二、输入/输出端口	170
三、I/O 端口的编址方式	171
四、接口电路应具备的功能	171
五、接口电路分类	172
六、8088 的输入/输出指令	172
§ 7.2 CPU 与输入/输出交换信息的方式	173
一、无条件传送方式	173
二、查询方式	174
三、中断控制方式	176
四、直接存储器存取(DMA)方式	177
§ 7.3 可编程定时器/计数器 8253	177

一、8253 的内部结构	177
二、8253 引脚功能	179
三、8253 控制字与初始化编程	180
四、8253 的工作方式	182
§ 7.4 8253 在 PC/XT 中的应用	188
§ 7.5 发声系统与音乐程序设计	190
一、PC 系列机发声系统	190
二、音符频率表	192
三、音乐程序设计举例	192
习题	196
第八章 中断系统	197
§ 8.1 中断的基本概念	197
§ 8.2 8088 的中断指令	198
§ 8.3 中断向量和中断向量表	199
§ 8.4 系统中断的分类	202
一、CPU 中断	202
二、软件中断	203
§ 8.5 8259A 中断控制器	205
一、8259A 的内部结构	205
二、中断响应周期	207
三、8259A 级连应用	207
四、8259A 中断管理方式	209
五、8259A 命令字	212
六、8259A 初始化编程	216
§ 8.6 PC/XT 硬件中断	216
一、8088 的中断引脚	217
二、PC/XT 可屏蔽中断	218
三、CPU 的中断响应过程	219
四、硬件中断和软件中断的区别	219
五、PC/XT 8259A 初始化	219
六、PC/XT 的非屏蔽中断	220
§ 8.7 PC/AT 硬件中断	220
一、PC/AT 可屏蔽中断	220
二、PC/AT 8259A 初始化	222
三、PC/AT 系统用户中断	223
§ 8.8 日时钟中断	223
一、日时钟中断处理流程	224
二、日时钟中断的外扩	224
§ 8.9 定时中断程序设计	224

一、中断类型的选择	224
二、中断向量的置换	225
三、定时中断程序设计	225
§ 8.10 实时时钟中断	232
一、实时时钟电路工作原理	233
二、实时时钟中断处理流程	233
三、周期中断	234
四、报警中断	237
§ 8.11 键盘中断	240
一、键盘中断全过程	240
二、PC/XT 机键盘接口电路	240
三、键盘中断处理流程	242
四、特殊键状态标志	242
五、键代码	243
六、键盘缓冲区	244
习题	245
第九章 串行通信与接口芯片	246
§ 9.1 串行通信基础	246
一、串行通信方式	246
二、串行异步通信协议	247
三、数据传送方式	248
四、信号的调制和解调	249
五、串行接口标准	249
六、串行通信接口的基本任务	250
§ 9.2 可编程串行异步通信接口芯片 8250	251
一、8250 内部结构	251
二、8250 的引脚特性	252
三、8250 的内部寄存器	255
四、8250 初始化编程	258
§ 9.3 可编程串行通信接口芯片 8251A	259
一、8251A 内部结构	259
二、8251A 工作原理	260
三、8251A 引脚功能	261
四、8251A 的命令字和状态字	263
§ 9.4 串行通信程序设计	265
一、BIOS 通信程序	265
二、串行通信的外部环境	268
三、串行通信程序设计	268
习题	273

第十章 并行接口芯片 8255A	274
§ 10.1 8255A 的内部结构	274
§ 10.2 8255A 控制字	277
§ 10.3 8255A 的工作方式	278
一、方式 0	279
二、方式 1	280
三、方式 2	283
§ 10.4 8255A 应用举例	285
习题.....	295
第十一章 DMA 控制器	296
§ 11.1 概述.....	296
§ 11.2 8237A DMA 控制器	297
一、8237A 的基本功能	297
二、8237A 的内部结构和工作原理	298
三、8237A 引脚功能	299
四、8237A 的内部寄存器	301
五、8237A 的时序	306
§ 11.3 8237A 的应用	307
一、8237A 的初始化编程	307
二、8237A 在 IBM PC/XT 中的应用	308
三、8237A 在 IBM PC/AT 中的应用	310
§ 11.4 DMA 传送实验	312
一、实验目的	312
二、实验电路	312
三、电路分析	312
四、实验内容	313
五、程序清单	314
习题.....	316
第十二章 数/模、模/数转换	317
§ 12.1 数/模转换	317
一、数/模转换原理	317
二、DAC 0832	318
三、数/模转换电路与程序设计	320
§ 12.2 模/数转换	323
一、模/数转换原理	323
二、ADC 0809	324
三、模/数转换电路	326

第十三章 汇编语言高级编程	327
§ 13.1 驻留程序	327
一、驻留程序的设计方法	327
二、驻留程序设计举例	330
三、DOS 内存管理	333
四、驻留程序的解驻	334
§ 13.2 音乐驻留程序	338
§ 13.3 时钟显示驻留程序	343
一、系统时钟的获取方法	343
二、系统时钟的显示方法	343
三、时钟显示驻留程序的激活方式	344
§ 13.4 菜单程序	349
一、菜单程序的设计考虑	349
二、常用的光标控制键	350
三、菜单程序设计举例	350
四、批处理程序的设计	356
§ 13.5 分屏幕显示全双工通信	357
§ 13.6 脉冲收号器性能模拟	365
一、拨号脉冲识别	365
二、脉冲收号扫描周期的确定	366
三、位间隔识别	366
四、位间隔扫描周期的确定	367
五、脉冲收号器性能模拟	367
第十四章 实验	375
§ 14.1 软件实验	375
§ 14.2 TPC—1 十六位微机实验培训系统简介	379
§ 14.3 硬件实验	382
参考文献	388

第一章 计算机中的数制和码制

§ 1.1 数 制

数制是研究数的表示方法。十进制计数是日常生活中广泛使用的进位计数制,而在计算机中,特别是计算机硬件,为了便于存储和运算,广泛采用二进制数。二进制数的每1位只有0、1两种数码,表示器件具有两种稳定状态,例如:触发器的置1、置0,开关的闭合与打开等均可表示1位二进制数。

由于二进制数书写格式冗长,不便阅读,在程序设计中又往往使用十六进制数、八进制数、二一十进制数等。

一、常用计数制

1. 十进制数

在程序设计中,广泛使用十进制数。十进制数的特点:每一位有0~9十种数码,因此基数为10,高位权是低位权的10倍,加减运算的法则为“逢十进一,借一当十”。

2. 二进制数

在计算机内部,一切信息均以二进制数形式出现。二进制数的特点:每一位仅有0、1两种数码,因此基数为2,高位权是低位权的2倍,加减运算的法则为“逢二进一,借一当二”。

3. 十六进制数

十六进制数是二进制数的另一种书写格式。由于二进制数书写格式冗长,不便阅读,人们把四位二进制数称为一组,每一组用等值的十六进制数表示(参阅表1.1.1)。十六进制数的特点:每一位有0~9和A~F十六种数码,因此基数为16,高位权是低位权的16倍,加减运算法则为“逢十六进一,借一当十六”。如:

$$(8A)_{16} + (59)_{16} = (E3)_{16}, (9C)_{16} - (6D)_{16} = (2F)_{16}$$

4. 八进制数

八进制数也是二进制数的另一种书写格式。把三位二进制数称做一组,每一组用等值的八进制数(实际上是十进制数中的0~7)表示。八进制数的特点:每一位有0~7八种数码,因此基数为8,高位权是低位权的8倍,加减运算法则为“逢八进一,借一当八”。

5. 二—十进制数

二—十进制数是用二进制编码表示的十进制数(Binary Coded Decimal——简称 BCD 码)。具体说就是用四位二进制数表示一位十进制数。如：

$$(5678)_{10} = (0101\ 0110\ 0111\ 1000)_{\text{BCD}}$$

任意进制的数,其等值的十进制数可以用下列的“按权展开式”表示。

$$N = D_{n-1} \times R^{n-1} + D_{n-2} \times R^{n-2} + \cdots + D_0 \times R^0 + D_{-1} \times R^{-1} + \cdots + D_{-m} \times R^{-m}$$

其中: n 为整数部分的位数, m 为小数部分的位数。

$D_{n-1} \sim D_0$ 为整数部分的每一位的数码, $D_{-1} \sim D_{-m}$ 为小数部分的每一位的数码。

$R^{n-1} \sim R^0$ 为整数部分的各数位的权值, $R^{-1} \sim R^{-m}$ 为小数部分的各数位的权值。

表 1.1.1 给出了四位二进制数和十进制数、十六进制数之间的等价关系。

表 1.1.1

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

二、数制换算

(1) 二进制数→十进制数,八进制数→十进制数,十六进制数→十进制数

根据按权展开式把每个数位上的代码和该数位的权值相乘,再求累加和即可得到等值的十进制数。如:

$$(1101.11)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (13.75)_{10}$$

$$(731)_8 = 7 \times 8^2 + 3 \times 8^1 + 1 \times 8^0 = (473)_{10}$$

$$(12A)_{16} = 1 \times 16^2 + 2 \times 16^1 + 10 \times 16^0 = (298)_{10}$$

(2) 二进制数→十六进制数

以小数点为界,四位二进制数为一组,每组用等值的十六进制数表示。如:

$$(1000111110.11)_2 = (23E.C)_{16}$$

(3) 十六进制数→二进制数

把十六制数的每一位用等值的四位二进制数代换。如:

$$(789A)_{16} = (0111\ 1000\ 1001\ 1010)_2$$

(4) 十进制数→二进制数

① 十进制整数→二进制数

运算方法为“除 2 取整”,即十进制整数被 2 除,取其余数,商再被 2 除,取其余数,……,

$N = (13)_{10} = (1101)_2$

运算过程： $13 \div 2 = 6$ 余数 $= 1 \cdots \cdots D_0$
 $6 \div 2 = 3$ 余数 $= 0 \cdots \cdots D_1$
 $3 \div 2 = 1$ 余数 $= 1 \cdots \cdots D_2$
 $1 \div 2 = 0$ 余数 $= 1 \cdots \cdots D_3$

所以 $N = D_3 D_2 D_1 D_0 = (1101)_2$

运算方法为“乘 2 取整”，即把十进制纯小数乘以 2，取其整数（不参加后继运算），乘积的小数部分再乘以 2，取整……，直到乘积的小数部分为 0。然后把每次乘积的整数部分按正序规律排列，即为等值的二进制数。如：

$$N = (0.6875)_{10} = (0.1011)_2$$

运算过程: $0.6875 \times 2 = 1.3750$	乘积的整数部分 = $1 \cdots \cdots D_{-1}$
$0.3750 \times 2 = 0.75$	乘积的整数部分 = $0 \cdots \cdots D_{-2}$
$0.75 \times 2 = 1.5$	乘积的整数部分 = $1 \cdots \cdots D_{-3}$
$0.50 \times 2 = 1.0$	乘积的整数部分 = $1 \cdots \cdots D_{-4}$

所以 $N = (0.1011)_2$

有些纯小数,不断地“乘 2 取整”也不能使其乘积的小数部分为 0,此时只能进行有限次运算,取其近似值。

整数部分用“除二取余”，小数部分用“乘二取整”分别运算，然后再行组合。

三、字符编码

数字、字母、符号统称字符。在微机系统中,字符的输入输出最常用的是美国信息交换标准代码,即 ASCII 码(American Standard Code for Information Interchange)。标准 ASCII 码,用 7 位二进制数做为字符的编码,但由于计算机通常用 8 位二进制数代表一个字节,故标准 ASCII 码也写成 8 位二进制数,但最高位 D_7 位恒为 0。 $D_6 \sim D_0$ 位代表字符的编码。表 1.1.2 为字符的 ASCII 码表。

表 1.1.2 **标准 ASCII 码字符表**

MSD LSD	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	,	P
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENG	NAK	%	5	E	U	e	u

续 表

MSD LSD	000	001	010	011	100	101	110	111
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	•	>	N	↑	n	~
1111	SI	VS	/	?	O	←	o	DEL

NUL	空	DLE	数据键换码
SOH	标题开始	DC1	设备控制 1
STX	正文开始	DC2	设备控制 2
ETX	正文结束	DC3	设备控制 3
EOT	传输结束	DC4	设备控制 4
ENG	询问	NAK	否定
ACK	认可	SYN	同步字符
BEL	报警(可听见声音)	ETB	信息组传送结束
BS	退一格	CAN	作废
HT	横向制表	EM	纸尽
LF	换行	SUB	减
VT	纵向制表	ESC	换码
FF	走纸控制	FS	文字分隔符
CR	回车	GS	组分隔符
SO	移位输出	RS	记录分隔符
SI	移位输入	VS	单元分隔符
SP	空格	DEL	删除

§ 1.2 码 制

在计算机中如何表示一个有符号的数？最常用的方法是：把数的最高一位定义为符号位，符号位为 0 表示正数，符号位为 1 表示负数，这样就把符号“数值化”了。有符号数的运算，其符号位上的 0 或 1 也被看作数值的一部分参加运算。

通常把用“+”、“-”表示的数称为真值数，把用符号位上的 0、1 表示正负的数称为机器数。机器数有三种表示方法。

一、机器数的原码、补码和反码

符号 ↓	符号位 ↓
设真值数 $X = +101$	原码机器数写成 $[X]_{\text{原}} = 0101$
$X = -101$	$[X]_{\text{原}} = 1101$
$X = +000$	$[X]_{\text{原}} = 0000$
$X = -000$	$[X]_{\text{原}} = 1000$
真值数 $X = +101$	反码机器数写成 $[X]_{\text{反}} = 0101$
$X = -101$	$[X]_{\text{反}} = 1010$
$X = +000$	$[X]_{\text{反}} = 0000$
$X = -000$	$[X]_{\text{反}} = 1111$
真值数 $X = +101$	补码机器数写成 $[X]_{\text{补}} = 0101$
$X = -101$	$[X]_{\text{补}} = 1011$
$X = +000$	$[X]_{\text{补}} = 0000$
$X = +0.101$	$[X]_{\text{补}} = 0.101$
$X = -0.101$	$[X]_{\text{补}} = 1.011$

由此可以得出结论：

第一、机器数比真值数多一个符号位。

第二、正数的原、补、反码和真值数相同。

第三、负数原码的数值部分与真值相同；

负数反码的数值部分为真值数按位取反；

负数补码的数值部分为真值数按位取反末位加一。

第四、没有负零的补码，或者说负零的补码和正零的补码相同。

第五、由于补码表示的机器数更适合运算，为此计算机系统中负数一律用补码表示。

第六、字长为 8 位的补码数，其真值范围是 $-128 \sim +127$ ；

字长为 16 位的补码数，其真值范围是 $-32768 \sim +32767$ 。

二、整数补码的运算

补码数怎样进行加减运算？为此先要引进几个概念：

1. 模

模是计量器的最大容量。一个 4 位的寄存器能够存放 $0000 \sim 1111$ 计 16 个数，因此它的模为 2^4 。一个 16 位的寄存器能够存放 $0 \cdots 0 \sim 1 \cdots 1$ ，共计 65536 个数，因此它的模为 2^{16} 。

2. 有模的运算

凡是用器件进行的运算都是有模的运算。例如利用 16 位的运算器，当运算结果超出