

第一章 计算机基础知识

本章要点：初步了解微型计算机硬、软件的系统组成；熟悉并掌握在计算机中的数据表示格式、数值运算方法和计算机程序算法的描述。

1.1 微型计算机系统组成

计算机系统由硬件系统和软件系统组成。硬件系统包括主机、存储器、输入设备（键盘、鼠标等）和输出设备（显示器、打印机等），如图 1.1 所示。主机主要由 CPU 即中央处理器（包括运算器、控制器）和内存（主存）组成，此外还包括时钟电路、中断控制电路、外存储器、DMA（直接存储器访问）电路、总线、附属电路等。其中 CPU 用于执行运算和控制整个计算机的工作。软件系统包括系统软件和应用软件。系统软件有操作系统、程序设计语言、编译系统、通用数据库管理系统等，应用软件有文字处理软件、表处理软件、应用数据库管理系统等。

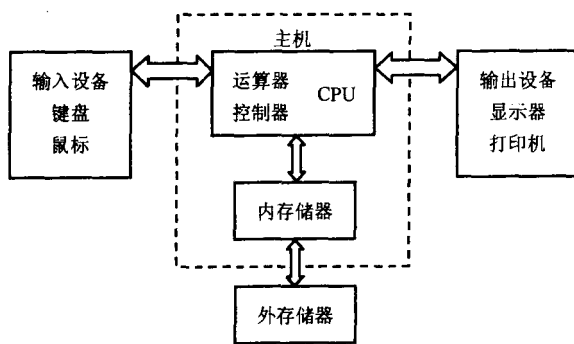


图 1.1 微型计算机硬件结构示意图

1.1.1 硬件系统

1. 中央处理器

中央处理器即 CPU，是计算机系统的核心部件，主要包括运算器和控制器。

(1) 运算器

运算器的英文简称为 ALU，它是按照特定的 CPU 指令来进行算术运算和逻辑运算的处理部件，通常由各种寄存器和累加器所组成。这些寄存器和累加器用来暂时保存一些数据和完成各种最基本的运算，其中包括算术运算和逻辑运算。运算器也可以在控制器的控制下，根据 CPU 指令的要求将处理结果送到存储器保存起来或通过特定的输出设备把结果显示出来。

(2) 控制器

控制器通常是由指令寄存器、指令计数器、译码器以及各种控制线路所组成。主要是用来完成对输入和输出的操作，以及对运算器起到控制作用。控制器可以从存储器中读取指令，经过译码器译码之后向计算机的各部件发出执行操作的控制信号，完成对该指令的执行。因此，控制器可以实现对输入和输出设备、存储器和运算器的控制和管理功能；可以将原始数据从输入设备或存储器中取出并送到运算器；也可以将运算器加工处理的结果送到存储器或输出设备中。

2. 存储器

存储器是计算机的记忆部件，用来存储计算机要进行信息处理的原始数据以及指示计算机如何工作的程序，并将运算的中间结果以及处理后的结果保存起来。

在微型计算机系统中，存储器能储存信息的能力称为存储容量。存储器中有大量的存储单元，每个存储单元都可以存储 8 位 (bit) 二进制信息。计算机系统中把能存储 8 位二进制信息的一个存储单元称为一个字节 (Byte, 简写 B)，它是存储容量的基本单位。把存储器中的每一个存储单元按顺序从 0 开始依次编号，这个编号称为地址。CPU 按照地址从存储器中存取数据。存储容量的单位除了字节外，还有 KB(千字节)、MB(兆字节)、GB(吉字节)、TB(太字节) 等，它们之间是以 1 024 为进位关系的。例如：1KB=1 024B；1MB=1 024KB；1GB=1 024MB；1TB=1 024GB。

从计算机使用者的观点来看，存储器的容量越大越好，存取信息的时间越短越好。Intel 8088CPU 能够提供 20 位的地址空间，它可以配置 1 MB 的存储器；Intel 80286CPU 能够提供 24 位的地址空间，它可以配置 16 MB 的存储器；Intel 80386CPU 能够提供 32 位的地址空间，它可以配置 4GB 的存储器；而 Intel 80486 和 PentiumCPU 能够提供 64 位的地址空间，所以它可以配置多达 16 千千太字节的存储器。

存储器分为内存储器和外存储器。内存储器又简称为内存或主存，它与计算机的 CPU 合在一起称为主机。

(1) 内存储器

内存储器用于存储运行中的程序和数据，它分为随机存取存储器（RAM）和只读存储器（ROM）两种。

从 RAM 中既可以读出数据又可以写入数据，但断电后 RAM 中不再保留原信息。因此，RAM 主要用来存放正在使用计算机的使用者的数据和程序。在计算机中还有一种 RAM 被称为高速缓冲存储器（Cache），它是同 CPU 一起工作，提高 CPU 与存储器或外部设备交换数据的速度的。

ROM 中每个单元的信息是固化了的，它是计算机制造厂家在机器出厂前用特殊方法写入的，使用者只能从中读取数据，而不能改变其中信息。这种存储器中的信息断电后不会丢失，只要通电后又可以从其中读出原来的信息。ROM 主要用来存放一些重要的且经常要使用的程序或数据，如计算机自身管理的系统程序等。ROM 还有其他两类：一类是可编程只读存储器（PROM），它用于计算机制造厂家一次性写入数据和程序，一旦写入数据和程序后，其内容是不能修改的；另一类是可擦除的可编程只读存储器（EPROM 和 EEPROM）。但是这种擦除必须使用特殊的工具或配有专门的擦除电路，在特定的情况下才能将原信息内容更改和擦除。在 Pentium 或笔记本计算机中都采用这类只读存储器存储计算机自身的系统管理程序，同时在机器中配有擦除电路，这样就使得系统管理程序很容易地得到升级，但同时也为计算机病毒提供了攻击的对象，有些计算机病毒专门攻击 EEPROM，使计算机彻底瘫痪。

（2）外存储器

外存储器又称外存或辅存（辅助存储器），位于主机的外部，存储容量一般都比较大，用来保存较大的程序或数据。常用的外存储器有软盘存储器、硬盘存储器、磁带存储器、光盘存储器等。

3. 输入、输出设备

输入和输出设备是实现人与计算机之间信息交换所必需的部件。

（1）输入设备

输入设备是外界向计算机传送数据或信息的装置，它负责将使用者要处理的数据及程序等信息输入到计算机存储器中。常用的输入设备有键盘、扫描仪、鼠标等。

（2）输出设备

输出设备是将计算机中的数据信息传送到机器外部，并转换成使人能够接受的表示形式。这些设备负责将处理后的结果或其他信息进行输出显示。常用的输出设备有显示器、打印机、绘图仪等。

4. 其他硬件电路

时钟电路：用于产生计算机工作时所必须的时钟控制信号。

中断控制电路：用于计算机中断过程的硬件控制。

DMA 电路：提供 DMA 过程的硬件控制。

总线：用于计算机中各个部件之间和计算机与外部设备之间的信息传递。

1.1.2 软件系统

软件和硬件一样都是计算机系统中的一个重要组成部分。软件系统是指能指挥计算机工作的程序以及程序在运行时所需要的各种数据的集合。微型计算机的软件系统可以分为系统软件和应用软件两大类。

1. 系统软件

系统软件是指计算机系统所必备的软件。它主要用来管理、监控以及维护计算机的各种软件和硬件资源。常用的系统软件有操作系统、语言处理程序、数据库管理系统以及各种工具软件等。

系统软件中最重要的是操作系统，操作系统是位于计算机硬件之上的最低层的系统软件，是对计算机硬件系统功能的首次扩充。所有其他系统软件和应用软件都是建立在操作系统的基础之上，它提供了使用者与计算机之间的一个接口。

目前使用最广泛的操作系统有 DOS（磁盘操作系统）、Windows（窗口操作系统）和 UNIX。其中 DOS 是单用户单任务的操作系统，是当前最为流行的标准微型计算机操作系统之一；而 Windows 和 UNIX 是应用最为广泛的多用户多任务操作系统，它们已成为 32 位高档微机及各种工作站的标准操作系统。尤其是多窗口、多用户、多任务的 Windows 操作系统，它给使用者提供了良好的界面形式，便于初学者进行操作，对于推动计算机的普及与应用起到了促进作用。

2. 应用软件

应用软件是根据使用计算机人员自己的需要，专门用于解决某个特定问题而编写的软件。它具有很强的针对性和实用性。随着计算机的不断推广和发展，各种应用软件也在各自的应用领域中发挥了越来越重要的作用。例如，信息管理软件、办公自动化系统、文字处理软件、计算机辅助设计及辅助教学软件、网络浏览、通信软件、多媒体视频和音频处理软件，等等，它们都显示出应用软件强大的生命力和广阔的发展前景。

12 计算机程序设计语言

语言可以分成两大类，即自然语言和人工语言。人类所使用的语言一般被

称为自然语言，它们是人类在自身发展过程中为了表达思想、交流经验、互通信息而自然形成的语言。人工语言是人们为了到达某种目的而专门设计的语言。

计算机的发明是人类科学技术发展历史上的一个伟大进步，计算机可以看成是人脑的延伸，但是目前它还不能理解人类的自然语言，只能接受特定的机器代码。要想使计算机按照人的意愿去处理某件事情，就必须通过某种途径来生成计算机所能够操作的代码。计算机语言就是用来生成这些可以为计算机直接识别的代码的人工语言之一。

计算机程序设计语言是用来编写程序的语言，它是软件系统的重要组成部分，与程序设计语言相对应的各种语言处理程序则为该语言提供支持和辅助作用。程序设计语言一般分为机器语言、汇编语言和高级语言 3 大类。

1.2.1 机器语言

计算机基本的物理构成是复杂的电子线路。这些线路可以在不同的电信号作用下，通过各种装置产生不同的动作。计算机的核心部分（CPU 存储器等）只能识别高、低电平两种状态，即二进制的 0 和 1。由 0 和 1 的不同组合所形成的可以为计算机直接识别的二进制指令代码的集合就称为机器语言。指令代码中一般包括两部分：操作码和地址码。操作码指出操作的性质，即告诉机器作何种操作；地址码则指出被操作的对象所在的位置。机器语言是一种能被计算机的硬件直接识别和执行的语言。计算机在接收到这些二进制代码后，由译码器产生各种控制信号送往有关的单元，从而产生相应的操作。

机器语言是最基本的、出现最早的计算机编程语言，是惟一可以为计算机直接执行的语言。用机器语言编写的程序冗余小、执行效率高、节省内存、运行速度快、可以直接控制计算机的硬件。但是不同类型计算机的硬件结构可能有所不同，它们各自具有不同的指令系统，因此用机器语言编程对程序设计者的水平要求很高，他们必须对所使用计算机的硬件工作原理以及线路连接关系十分清楚，而且一般在某种机器上编写的程序不能在另一种机器上运行，因此人们称机器语言为“面向机器”的程序设计语言。此外，用机器语言编写的程序可读性差、可移植性差、容易出错，出错后很难查找，因此，人们往往不直接使用机器语言进行编程，而是借助于其他更为简单、编程效率更高的计算机语言，通过间接的方法来产生可以为计算机直接接受的代码。

1.2.2 汇编语言

为了克服机器语言的缺点，便于一般人员进行编程，在 20 世纪 50 年代初

出现了汇编语言。其特点是引入了一些助记符来表示操作，用一些特定的符号来表示计算机中的某些单元，例如，用“ADD”表示加运算，用“AL”、“AH”表示某一寄存器，等等。这些助记符比较接近英文含义，也比较简单，便于记忆、理解，从而简化了编程过程。由于汇编语言比机器语言更直观，所以使用汇编语言编写的程序比用机器语言编写的程序更易读和易修改。

使用助记符和有关符号编写的程序称为汇编语言程序。由于计算机只能识别二进制代码，不能识别这些符号，因此，还必须通过某种方法将汇编语言程序“翻译”成相应的由二进制代码组成的目标程序。将汇编语言程序进行“翻译”的过程称为汇编，能完成汇编任务的软件叫汇编程序。

汇编语言同机器语言一样，仍然是一种属于“面向机器”的程序设计语言，编程过程离不开机器的硬件结构，助记符、符号及汇编过程都要依赖于机器的特定指令系统。所以，汇编语言程序的通用性仍然较差，应用范围受到一定的限制。但是，汇编语言程序具有运行速度快、能直接控制硬件等特点，在实时性要求较高的计算机自动控制系统中或者对计算机硬件设备要求特殊管理、编写系统软件的场合下，仍然得到了广泛的应用。

1.2.3 高级语言

机器语言和汇编语言都是“面向机器”的程序设计语言，人们习惯上称它们为“低级语言”。由于它们难学、难记、程序难调试、可移植性差，所以在非计算机专业人员中不易推广。随着计算机的迅速普及和人们解决日益增加的实际问题的需要，出现了各种形式的高级语言，高级语言又称为算法语言，它是一种面向问题的程序设计语言。

高级语言的各种命令的形式接近于自然语言（英语）和数学算式的格式表示。它们有着各自的特点和各自严格的语法规则，便于记忆、书写、阅读和修改。使用高级语言编写出的程序的每一条命令，从字面上就能看出其含义。高级语言基本上摆脱了机器类型的影响，程序设计者在进行程序设计时可以不考虑机器的硬件结构，只需要掌握应用问题的解决方法和有关的算法，按照语言的语法规则书写命令，就可以编出程序。

使用某种高级语言编写出来的程序称为该语言的源程序。计算机不能直接辨认用高级语言编写的程序中的指令，必须将高级语言程序“翻译”成计算机可以直接辨认的机器语言程序。然而，用人工进行这样的“翻译”，在实际上是不可能的。因此，人们在创造高级语言的时候，同时要编写出用计算机将高级语言程序“翻译”成机器语言程序的软件，这样的“翻译”软件就叫做高级语言的编译软件（程序）。

使用高级语言编程的方法和思路很接近人与人之间的自然语言交流和数学

描述，因此编程效率高、编程的时候基本上不涉及计算机的硬件知识、便于普及、程序的通用性好。但是高级语言程序不如机器语言简练、翻译转换后生成的目标程序冗余大、运行时占用内存多、速度较慢。

用高级语言编写程序时与具体的计算机硬件无关，因此可大大简化程序的编制和调试工作，而且用高级语言编写的程序通用性强、可移植性好。例如，用 Pascal、C语言等高级程序设计语言所编写的程序就是典型的代表。

1.2.4 编程工具软件

编程工具软件是为开发、研制、跟踪或调试其他程序或软件提供服务的 一类软件。这类软件为使用者编制程序、开发软件以及灵活使用计算机提供了一种界面和窗口，极大地方便了编程人员。这类软件主要有：

(1) 编辑程序：主要用于源程序的录入或修改。它是计算机系统中必不可少的工具软件之一。

(2) 编译程序：主要用于将源程序编译成可执行的机器代码程序。指定计算机 CPU 的机器代码。

(3) 调试程序：调试程序主要用于对新编制的程序进行调试，用以发现或寻找错误。

1.3 计算机运算的基础知识

在日常生活和学习中，人们常常习惯于用十进制进行计数。十进制计数的特点是“逢十进一”，且需用到 0~9 共 10 个数码。但对于计算机来说，它采用的是二进制计数法，计算机中的数之所以采用二进制计数法，是因为数在计算机中是以电子器件的物理状态来描述的，比如，用两种不同的状态低电平和高电平来表示二进制的两个数字符号 0 和 1，这样的运算电路较容易实现。在计算机中所有运算和判断都是通过二进制来体现的，从数据到指令、从地址到内容都是用二进制来实现的。但二进制书写起来太长、不方便，因此，在计算机数值体系中也常用八进制或十六进制数来表示。八进制或十六进制与二进制之间有着很方便的换算关系。

无论对于哪种进制形式，都包含两个基本要素：基数和位权。基数是指该进制中允许使用的数码个数。比如十进制中允许使用 0~9 共 10 个数码，故十进制的基数为 10；位权是指以该进制的基数为底，数码所在位置的序号为指数的整数次幂。比如十进制的位权为 10^k (k 为整数)，把十进制数 123.45

按位权展开后为（下标表示数制）：

$$(123.45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

表 1.1 列出了数制、基数、数制的规则、数码的个数之间的对应关系。

表 1.1 数制、基数、数制的规则、数码的个数

数制	基数	数制的规则	数 码
二进制	2	逢二进一	0,1
八进制	8	逢八进一	0,1,2,3,4,5,6,7
十进制	10	逢十进一	0,1,2,3,4,5,6,7,8,9
十六进制	16	逢十六进一	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

1.3.1 计算机系统使用的数制及其相互转换

1. 二进制数

二进制数中只有两个数码 0 和 1，其计数特点及进位规则是“逢二进一”。二进制的基数为 2，位权为 2^k （ k 为整数），例如，二进制数 1010 按位权展开为：

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

将二进制数转换成十进制数时，只要按权展开相加即可。为了与十进制数相区别，常在二进制数后面写上字母 B，在十进制数后面写上字母 D。例如：1010B、2345D……

由于二进制数只取两个数码 0 和 1，因此，二进制数的每一位都可以用任何两个不同稳定状态来表示，比如电灯的亮和灭、晶体管的截止和导通、脉冲的有无、电位的高低等。只要规定其中的一种状态表示“1”，另一种状态表示“0”，就可以用它们来表示二进制了。

2. 八进制数

八进制数中共有 0~7 共 8 个数码，其计数特点及进位规则是“逢八进一”。八进制的基数为 8，位权为 8^k （ k 为整数）。

八进制以 8 为基数，每一位都有 8 个状态，即用数码 0, 1, 2, 3, 4, 5, 6, 7 来表示，恰好 3 位二进制数可以对应 1 位八进制数。八进制数后面用字母“O”作标记。例如，八进制数 1234 按位权展开为：

$$(1234)_8 = 1 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 4 \times 8^0$$

将八进制数转换成十进制数时，也是只要按位权展开相加即可。

3. 十六进制数

十六进制数中有 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 以及 A, B, C, D, E, F 共 16 个数码，其计数特点及进位规则是“逢十六进一”。十六进制的基数

为 16^k ,位权为 16^k (k 为整数)。例如,十六进制数 1234 按位权展开为:

$$(1234)_{16} = 1 \times 16^3 + 2 \times 16^2 + 3 \times 16^1 + 4 \times 16^0$$

将十六进制数转换成十进制数时,同样是只要按位权展开相加即可。

由于二进制书写起来很长,读起来不易懂、不方便,因此,在编写程序过程中常用十六进制数来表示。十六进制数后面用字母“H”作标记。十六进制数的每一个数位都有十六个状态,即用数码 0~9 和 A~F 来表示,恰好 4 位二进制数对应 1 位十六进制数,其中 A~F 表示十进制数的 10~15。

在汇编语言中,使用十六进制数时,有一点需要注意:凡是以字母 A~F 打头的十六进制数,都要以数字“0”作为开头,以避免与标识符和寄存器名相混淆。例如,十六进制数 B0H 应写成 0B0H。

4. 常用数制对照表

表 1.2 列出了十进制、二进制、十六进制之间的对应关系。

表 1.2 十进制、二进制、十六进制之间对应关系

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

5. 各进位数制间的转换

在计算机内部处理的数据都是二进制数,但编程时往往为了使用上的方便,需要用到一些其他进制的数据。因此,需要掌握各进制数之间的转换方法。

(1) 二进制转换成十进制

将二进制的各位按位权展开后相加即可得到相应的十进制数。

$$\begin{aligned} \text{例 1.1 } (1010)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 8 + 0 + 2 + 0 \\ &= (10)_{10} \end{aligned}$$

(2) 十进制数转换成二进制数

将一个十进制数转换成二进制数时,需将整数部分和小数部分分开分别进行转换。十进制整数转换为二进制整数时采用“除 2 取余法”,即将十进制数除以 2,得到一个商数和一个余数;再将商数除以 2,又得到一个商数和一个

余数.....继续这个过程，直到商数为 0 时为止。最后将每次所得到的余数（0 或 1）按逆序排列得到相应的二进制数。

例 1.2 将十进制数 53 转换为二进制数。

转换过程为： $53 \div 2 = 26$ （余 1）——（二进制数最低位）
 $26 \div 2 = 13$ （余 0）
 $13 \div 2 = 6$ （余 1）
 $6 \div 2 = 3$ （余 0）
 $3 \div 2 = 1$ （余 1）
 $1 \div 2 = 0$ （余 1）——（二进制数最高位）

最后结果为 $(53)_{10} = (110101)_2$

十进制小数转换成二进制数采用“乘 2 取整法”，即用 2 乘十进制小数，得到一个整数部分和一个小数部分；再用 2 乘小数部分，又得到一个整数部分和一个小数部分.....继续这个过程，直到余下的小数部分为 0 或者满足精度要求为止。最后将每次得到的整数部分（0 或 1）从左到右排列即可得到所对应的二进制小数。

例 1.3 将十进制小数 0.65 转换为二进制小数。

转换过程为： $0.65 \times 2 = 1.3$ （整数部分为 1 小数部分 3）
 $0.3 \times 2 = 0.6$ （整数部分为 0 小数部分 6）
 $0.6 \times 2 = 1.2$ （整数部分为 1 小数部分 2）
 $0.2 \times 2 = 0.4$ （整数部分为 0 小数部分 4）
（根据小数精度要求终止过程）

最后结果为 $(0.65)_{10} = (0.101\cdots)_2$

（3）十进制数转换成十六进制数

同十进制数转换成二进制数一样，在将一个十进制数转换成十六进制数时，需要将整数部分和小数部分分开单独进行转换。整数转换采用“除 16 取余法”，小数转换采用“乘 16 取整法”。

例 1.4 将十进制数 53.75 转换成十六进制数。

转换过程为：

整数部分 $53 \div 16 = 3$ （余 5）——（十六进制数最低位）
 $3 \div 16 = 0$ （余 3）——（十六进制数最高位）

整数部分结果为 $(53)_{10} = (35)_{16}$

小数部分 $0.75 \times 16 = 12$ （整数部分为 12 小数部分为 0）
 $(12)_{10} = (0C)_{16}$

小数部分结果为 $(0.75)_{10} = (0.C)_{16}$

最终结果为 $(53.75)_{10} = (35.C)_{16}$

(4) 二进制、八进制与十六进制之间的转换

由于 2 的 3 次方等于 8, 2 的 4 次方等于 16, 所以 3 位二进制数相当于 1 位八进制数, 而 1 位 16 进制数相当于 4 位二进制数。

八进制转换成二进制的方法为: 每位八进制数用相应的 3 位二进制数代替。

二进制转换成八进制的方法为: 从小数点开始, 向前每 3 位一组构成 1 位八进制数; 向后每 3 位一组构成 1 位八进制数, 当最后一组不够 3 位时, 在后面添 0 补足 3 位。

十六进制转换成二进制的方法为: 每位十六进制数用相应的 4 位二进制数代替。

二进制转换成十六进制的方法为: 从小数点开始, 向前每 4 位一组构成一位十六进制数; 向后每 4 位一组构成 1 位十六进制数, 当最后一组不够 4 位时, 在后面添 0 补足 4 位。

例 1.5 将八进制数 123.45 转换成二进制数。

1	2	3	.	4	5
↓	↓	↓		↓	↓
001	010	011	.	100	101

最后结果为 $123.45_{10} = 1010011.100101_{2}$

例 1.6 将十六进制数 5B.3E 转换成二进制数。

5	B	.	3	E
↓	↓		↓	↓
0101	1011	.	0011	1110

最后结果为: $5B.3E_{16} = 1011011.0011111_{2}$

例 1.7 将二进制数 101011001101.010011001 分别转换成八进制数和十六进制数。

转换成八进制:	101	011	001	101	.	010	011	001
	↓	↓	↓	↓		↓	↓	↓
	5	3	1	5	.	2	3	1

最后结果为 $101011001101.010011001_{2} = 5315.231_{8}$

转换成十六进制:	1010	1100	1101	.	0100	1100	1000
	↓	↓	↓		↓	↓	↓
	A	C	D	.	4	C	8

最后结果为 $101011001101.010011001_{2} = ACD.4C8_{16}$

1.3.2 二进制数的算术运算和逻辑运算

1. 算术运算

计算机中用二进制数的形式表示信息。算术运算按二进制运算规则进行。二进制运算规则如下所述：

规则 1 加法运算

$0+0=0$; $0+1=1$; $1+0=1$; $1+1=10$ (有进位 1)

规则 2 减法运算

$0-0=0$; $0-1=1$ (向高位借 1); $1-0=1$; $1-1=0$

规则 3 乘法运算

$0 \times 0=0$; $0 \times 1=0$; $1 \times 0=0$; $1 \times 1=1$

例 1.8 $1101\text{B}+11\text{B}$

$$\begin{array}{r} 01101 \\ +) 00011 \\ \hline 10000 \end{array}$$

例 1.9 $1101\text{B}-11\text{B}$

$$\begin{array}{r} 1101 \\ -) 0011 \\ \hline 1010 \end{array}$$

例 1.10 $1101\text{B} \times 11\text{B}$

$$\begin{array}{r} 1101 \\ \times) 11 \\ \hline 1101 \\ +) 1101 \\ \hline 100111 \end{array}$$

2. 逻辑运算

计算机中用两种状态表示二进制的 0 和 1,也可以表示逻辑值的“是”(“T” — True)与“非”(“F” — False)。对逻辑值的运算称为逻辑运算。逻辑运算的一个特点是按位进行运算,即逻辑运算中没有进位问题。一般用二进制的“1”表示逻辑的“是”或“真”(“T”),用二进制的“0”表示逻辑的“非”或“假”(“F”)。

逻辑运算可以和普通代数一样使用字母表示变量,称为逻辑变量。逻辑变量的取值只有两种,“1”或“0”。

在逻辑代数中有 4 种常用运算规则和 9 个基本定律:

规则 4 “乘”运算。或称“与”运算,运算符号为“ \cdot ”或者“ \wedge ”。

$$0 \cdot 0 = 0; 0 \cdot 1 = 0; 1 \cdot 0 = 0; 1 \cdot 1 = 1$$

当使用逻辑变量进行“与”运算时， $A \cdot B$ 可以简写成 AB 。

规则 5 “和”运算。或称“或”运算，运算符号为“+”或者“ \vee ”。

$$0+0=0; 0+1=1; 1+0=1; 1+1=1$$

当使用逻辑变量进行“或”运算时，表达式为 $A+B$ 或者 $A \vee B$ 。

规则 6 “非”运算。或称“反”运算，运算符号是在逻辑值或变量符号上加“-”。

$$0=1; 1=0$$

当使用逻辑变量进行“非”运算时，表达式为 A 。

规则 7 “异或”运算。运算符号为“ \oplus ”。

$$0 \oplus 0 = 0; 0 \oplus 1 = 1; 1 \oplus 0 = 1; 1 \oplus 1 = 0$$

当使用逻辑变量进行“异或”运算时，表达式为 $A \oplus B$ ，它等价于 $A \cdot B + A \cdot \bar{B}$ 。

例 1.11 若 $A=1, B=0$ 。求 A 与 B 、 A 或 B 、 A 异或 B 和 A 非、 B 非。

$$A \cdot B = 0; A + B = 1; A \oplus B = 1; \bar{A} = 0; \bar{B} = 1$$

定律 1 0, 1 律

$$A \cdot 0 = 0; A \cdot 1 = A; A + 0 = A; A + 1 = 1$$

定律 2 交换律

$$A + B = B + A; A \cdot B = B \cdot A$$

定律 3 结合律

$$A + B + C = (A + B) + C = A + (B + C)$$

$$A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

定律 4 分配律

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

定律 5 重叠律

$$A + A = A; A + A + A + \dots + A = A; A \cdot A = A; A \cdot A \cdot A \cdot \dots \cdot A = A$$

定律 6 互补律

$$A + \bar{A} = 1; A \cdot \bar{A} = 0$$

定律 7 吸收律

$$A + A \cdot B = A; A \cdot (A + B) = A; A + A \cdot B = A + B; A \cdot (A + B) = A \cdot B$$

定律 8 对合律

$$\bar{\bar{A}} = A$$

定律 9 摩根定理

$$\overline{(A + B)} = \bar{A} \cdot \bar{B}; \overline{A \cdot B} = \bar{A} + \bar{B}$$

在计算机程序设计语言设计中，经常需要使用逻辑运算（或称为布尔逻辑运算）

对数据进行处理。最常用的布尔逻辑运算就是“与”、“或”、“非”、“异或”运算以及运用逻辑运算定律对逻辑表达式进行简化处理。

例 1.12 设变量 A 的内容是 8 位二进制数，即 $d_7d_6d_5d_4d_3d_2d_1d_0$ ，其中 d_7 为最高位。

(1) 将变量 A 的内容全部清“0”： $A \cdot (00000000)$ 或者 $A \oplus A$ 。

说明：一个变量自身进行异或操作，其结果是把该变量内容清“0”。

(2) 将变量 A 的内容中 d_4 位清“0”，其他位不变： $A \cdot (11101111)$ 。

(3) 将变量 A 的内容中 d_4 位置“1”，其他位不变： $A + (00010000)$ 。

(4) 将变量 A 的内容中各位一律求反： A 或者 $A \oplus (11111111)$ 。

说明：一个二进制数各位与“1”相“异或”，其结果是把它求反。

1.3.3 计算机中数据和字符的表示形式

1. 有符号的二进制数的表示

前面所涉及到的二进制数均为不带正 (+)、负 (-) 号的数，不管这个数含有多少个二进制数位，这些数都是该数的一个组成部分。例如，在计算机中，一个 16 位无符号数的表示形式为： $d_{15}d_{14}d_{13}d_{12}d_{11}d_{10}d_9d_8d_7d_6d_5d_4d_3d_2d_1d_0$ 。它能表示数的最小值为 0000H，最大值为 0FFFFH，也就是说，它能表示数的范围为 0H~0FFFFH，转换成十进制数为 0~65 535。

在计算机中，数的符号也是用二进制来表示的，把一个数连同其符号一起在机器中表示称为机器数。在这里最高位为“0”表示正数，最高位为“1”表示负数，即把最高位 (d_{15}) 作为该数的符号位。

带符号数在计算机中的表示，通常是利用该数的最高有效二进制位来作为其符号位。由于符号位占据了一个二进制位，所以，有符号的二进制数与相应位数的无符号二进制数相比，其表示的数值范围有所不同。例如，一个 16 位有符号数表示形式为： d_{15} (符号位) $d_{14} \sim d_0$ 。它能表示数的最小值为 8000H，最大值为 7FFFH，即数的取值范围为 8000H~7FFFH，转换成十进制数为 -32 768~32 767。

2. 数的补码表示

在计算机内部，对任意一个带符号的二进制数来说，都是按其补码的形式来进行存储和处理的。一个带符号数的补码的计算方法为：正数的补码与其原二进制数相同；而负数的补码形式可以先写出该负数所对应正数的二进制形式，然后使该正数的每一个二进制位取反（即 0 变 1，1 变 0），最后再将取反后的结果末位加 1，从而得到该负数所对应的补码表示形式。

在计算机中，实际上只有加法运算。减法运算是转换成加法运算来进行的。同样，乘法运算也是转换成加法运算进行的，除法运算是转换成减法运算进行

的。一般在计算机的运算中是用补码进行加减运算的。在用补码进行加减运算时，符号位是一起参加运算的。

例 1.13 求 -7 的补码表示形式。

-7 所对应正数的二进制形式为 00000111 每一位取反后为 11111000，末位加 1，即 $11111000+00000001=11111001$ 。

-7 的补码表示形式为：11111001B，十六进制表示形式为：0F9H。

因此，在计算机内部，-7 是以 11111001B 的形式来参加运算或存储的。

例 1.14 (8-7) 的计算机运算形式。

$$8-7=8+(-7)=00001000B+11111001B=00000001B=1$$

如果用十六进制数表示+1，则其在计算机内部的补码表示形式为 0001H，而-1在计算机内部的补码表示形式为 0FFFFH。

3. 数值的存储形式

数值在计算机中的存储主要有以下几种形式：位、字节、字、双字、四字、十字节和字符串。

(1) 位(bit)

位是由单一的 1 位二进制数所构成的。在计算机中，位是数的最小表示形式，每一位代表一种状态，而这种状态只能是 0 或 1。

(2) 字节(Byte)

由 8 位连续的二进制位组成 1 个字节。在计算机内部，字节是存储单元的最基本单位。每个存储单元中存放的数据称为该存储单元的内容，这些内容可以是字符、数字、计算机指令或另一个存储单元的地址等信息。

字节的最低位称为第 0 位，最高位称为第 7 位，对于无符号整数来说，它可以存储的最小数值为 00000000B 最大数值为 11111111B(255D)，即在一个字节的存储单元中可以存储 0~255 之间的任意一个正整数。

(3) 字(Word)

由两个连续字节组成一个字，即字由 16 个二进制位组成。并且将低 8 位称为低有效字节，将高 8 位称为高有效字节。当把一个整数字保存在存储器中时，这两个字节的存放顺序正好是相反的：高有效字节存储在高地址部分，低有效字节存储在低地址部分。例如存储 1234H 这个数时，其存放顺序如图 1.2 所示。

(4) 双字(DWord)

双字由两个连续存放的相邻的字组成(4 个字节)，其宽度为 32 位。双字的存储顺序与字是一样的。利用双字进行运算的一大优点是可以提高运算的精度。

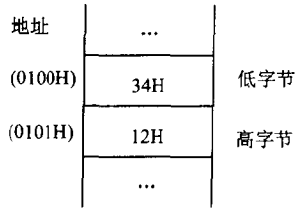


图 1.2 计算机存储单元的地址和内容

(5) 四字 (QWord)

四字由 4 个连续字所组成,其宽度为 64 位。它可以解决双字满足不了的精度问题。其存储结构与前相似。

(6) 十字节 (TByte)

十字节是由 10 个连续字节所组成,其宽度为 80 个二进制位,可用来存储极大的数字或字符串。

(7) 字符串

字符串又称串,它是用单引号引起来的能表示多个字节数据的完整数据。

例如: 'This is a string!'就是一个字符串。在字符串中,每一个字符在存储时都占用 1 个字节,将字符串所占用的字节总数称为该字符串的长度。例如,字符串'This is a string!'的长度为 17 即它在内存中存储时所占用的字节数为 17。

4. 在计算机中的常用编码

(1) ASCII 码 (字符编码)

在计算机中,不仅数值数据是用二进制来表示的,其他一些数据如汉字和符号等非数值数据,也是以二进制编码的形式存放在机器的内部。为了便于信息的表示、存储和处理,人们规定了多种标准的编码方式,目前在微型机上广泛使用的 ASCII 码是其中最典型的一种编码形式。

ASCII (American Standard Code for Information Interchange) 码又称美国标准信息交换代码,它已被国际标准化组织 (ISO) 定为国际标准。ASCII 码是能够表示字母、数字、专用字符和控制字符 4 种类型的 7 位编码,共能表示 128 个字符。其中包括 10 个阿拉伯数字、52 个英文大小写字母、33 个符号和运算符以及 33 个控制符。每个字符长度为 7 位,最高位常用于奇偶校验。

在 ASCII 码表中,字符的排列有一定的规律性,例如,数字 0~9 编码的高 3 位均为 011 低 4 位与其对应数值的二进制代码相同字母 A~Z 编码从 41H 连续递增到 5AH。采用这种编码方式便于进行信息的检索。ASCII 码表见附录一。

(2) BCD 码

BCD 码又称为无符号十进制数。它是用二进制编码来表示的十进制数。

根据其存储方式不同，BCD 码可以划分为两种类型：压缩的 BCD 码（又称无符号压缩十进制数）和非压缩的 BCD 码（又称无符号非压缩十进制数）。

压缩的 BCD 码是用 1 个字节来表示两位二进制编码的十进制数，而每位十进制数用 4 位二进制数来表示，高 4 位表示十进制数的十位数字 0~9，低 4 位表示十进制数的个位数字 0~9。其表示十进制数的范围为 0~99。例如，十进制数 56，用压缩 BCD 码表示形式为 01010110B。

非压缩的 BCD 码是用两个字节来表示两位二进制编码的十进制数，每位十进制数用 1 个字节来表示，第一个字节用来表示十位数字，第二个字节用来表示个位数字。

非压缩的 BCD 码其数值放在每个字节的低 4 位中。在通常情况下，高 4 位数值都用 0 来表示。但对于加减运算来说，由于高 4 位不影响运算结果，其值可以是任意的。

两个字节的非压缩 BCD 码可表示的十进制数范围同样也是 0~99。例如，56 的非压缩 BCD 码表示形式为 0000010100000110B。

（3）汉字编码

汉字信息交换码是用两个 ASCII 码代表一个汉字编码。这是我国国家标准《信息交换用汉字编码字符集——基本集》，其代号为 GB 2312-80，通常称作国标码。为了与 ASCII 码相区别，汉字的两个 ASCII 码前都加一个“1”，因此一个汉字由两个字节组成，最高位均为“1”。

1.4 计算机程序算法的表述

要想让计算机完成某件任务，首先需要分析这个任务，确定完成各个步骤，然后选择一种计算机语言将这些步骤表示出来，这就是程序设计。程序最终以二进制机器语言的形式为计算机所接受，从而产生一系列相应的操作。为了有效地进行程序设计，一方面要掌握解决问题（包括数值问题和非数值问题）的方法和步骤，另一方面还要能熟练地用某种程序设计语言将其正确地表示出来。

程序设计语言只是一种工具，用来表示处理问题的方法和步骤。只学会语言而不会设计解题的方法和步骤，还是不能解决问题。可以说，程序设计的灵魂是算法，而语言只是形式。有了正确的算法，就可以利用任何一种语言编写程序，使计算机进行工作，得出正确的结果。