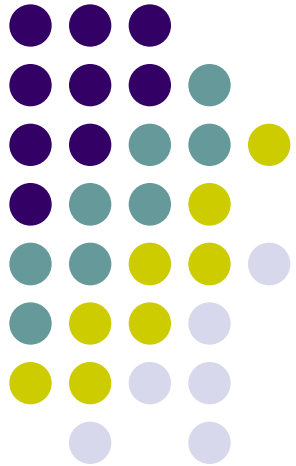
计算机基础概述

计算机中信息的表示

张华



计算机的二进制世界

计算机的世界是一个二进制世界

计算机是信息处理的工具，任何信息必须被转换成二进制形式数据后才能由计算机进行处理、存储和传输。

常用的数制

数制

数制也称进位计数制，是指用一组固定的符号和统一的规则来表示数值的计数方法。

特点：表示数值大小的数码与它在数中的位置有关。

例如：十进制，二进制，八进制，十六进制

十进制数

十进制数

十进制数是由0、1、2、3、4、5、6、7、8、9十个不同的符号组成，每一个符号处于数中不同的位置时，它所代表的实际数值是不一样的。

例如：1999 可表示成：

$1\times1000+9\times100+9\times10+9\times1$

$=1\times10^3+9\times10^2+9\times10^1+9\times10^0$

每个数字符号的位置不同，它所代表的数值也不同，这就是经常所说的个位、十位、百位、千位……的意思。

二进制数

二进制数

二进制数包括两个数字符号：0和1。数中0和1的位置不同，它所代表的数值也不同。

例如：二进制数1101表示十进制数13。

$$(1101)_2 = 1\times2^3+1\times2^2+0\times2^1+1\times2^0$$

$$= 8+4+0+1 = 13$$

二进制数的基本特点

- 两个数字符号：0和1
- 逢二进一

不同进制数的表示方法

一般用（　　）角标表示不同进制的数。

例如：十进制数用（　　）$_{10}$表示，二进制数用（　　）$_{2}$表示

（100）$_{10}$ 表示十进制数 100

（100）$_{2}$ 表示二进制数 100

※在微机中，一般在数字的后面，用特定字母表示该数的进制。

B-二进制	1010B 是二进制数 1010
D-十进制（D可省略）	1010D或1010 是十进制数 1010
O-八进制	1010O 是八进制数 1010
H-十六进制	1010H 是十六进制数 1010

进位计数制的三要素

在进位计数制中有数位、基数和位权三个要素。

◆数位是指数码在一个数中所处的位置。

◆基数是指在某种进位计数制中，每个数位上所能使用的数码的个数。

- 二进制数基数是2
- 每个数位上所能使用的数码为0和1二个数码。

◆对于多位数，处在某一位上的“1”所表示的数值的大小，称为该位的位权。

- 二进制第2位的位权为2，第3位的位权为4。
- 一般情况下，对于N进制数，整数部分第i位的位权为N^{i-1}，而小数部分第j位的位权为N^{-j}。

※在数制中有一个规则：如果是N进制数，必须是逢N进1。

十进制

十进制（十进位计数制）

- 具有十个不同的数码符号0、1、2、3、4、5、6、7、8、9
- 基数为10
- 逢十进一

$$(1011)_{10} = 1\times10^3+0\times10^2+1\times10^1+1\times10^0 = 1011$$

按权展开求和

值

二进制

二进制（二进位计数制）

- 具有两个不同的数码符号0、1
- 基数为2
- 逢二进一

$$(1011)_2 = 1\times2^3+0\times2^2+1\times2^1+1\times2^0 = (11)_{10}$$

八进制

八进制（八进位计数制）

- 具有八个不同的数码符号0、1、2、3、4、5、6、7
- 基数为8
- 逢八进一

$$(1011)_8 = 1\times8^3+0\times8^2+1\times8^1+1\times8^0 = (521)_{10}$$

十六进制

十六进制（十六进位计数制）

- 具有十六个不同的数码符号0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F
- 基数为16
- 逢十六进一

$$(1011)_{16} = 1\times16^3+0\times16^2+1\times16^1+1\times16^0 = (4113)_{10}$$

四位二进制数与其它进制数的对应表

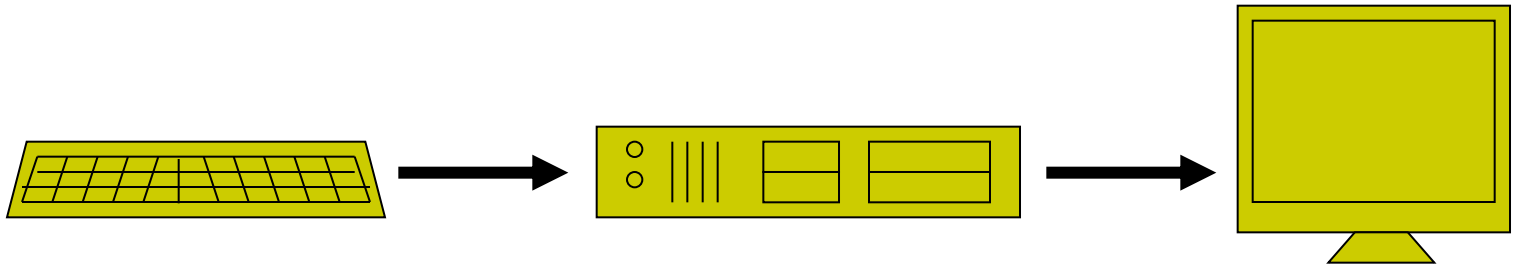
二进制	十进制	八进制	十六进制
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000		10	8
1001			9
1010			A
1011			B
1100			C
1101			D
1110	14	16	E
1111	15	17	F

别混淆了数与数值

不同进制数之间的转换

用计算机处理十进制数，必须先把它转化成二进制数才能被计算机所接受；

同理，应将计算结果的二进制数转换成人们习惯的十进制数。



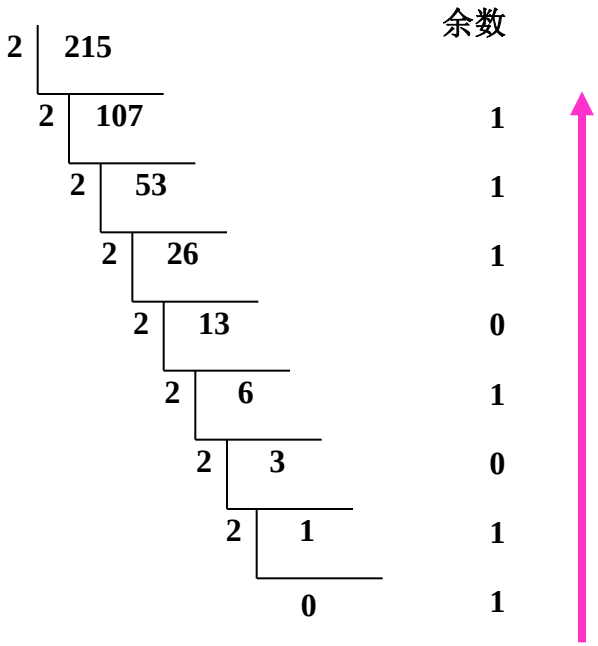
输入：十进制　　处理：二进制　　输出：十进制

十进制整数转换成二进制整数

“除2取余法”

把被转换的十进制整数反复地除以2，直到商为0，所得的余数（从末位读起）就是这个数的二进制表示。

将十进制整数（215）$_{10}$转换成二进制整数



（215）$_{10}$＝（11010111）$_2$

十进制整数转换成其它进制整数

十进制整数转换成八进制整数的方法是“除8取余法”；

十进制整数转换成十六进制整数的方法是“除16取余法”。

十进制小数转换成二进制小数

“乘2取整法”

将十进制小数连续乘以2，选取进位整数，直到满足精度要求为止 。

将十进制小数（0.6875）$_{10}$转换成二进制小数

0.6875
×） 2
1.3750　　整数＝1
0.3750
×） 2
0.7500　　整数＝0
×） 2
1.5000　　整数＝1
0.5000
×） 2
1.0　　整数＝1

（0.6875）$_{10}$＝（0.1011）$_{2}$

十进制小数转换成其它进制小数

十进制小数转换成八进制小数的方法是“乘8取整法”；

十进制小数转换成十六进制小数的方法是“乘16取整法”。

二进制数转换成十进制数

将二进制数按权展开求和

将（10110011.101）$_2$ 转换成十进制数

1×2^7 代表十进制数128

0×2^6 代表十进制数0

1×2^5 代表十进制数32

1×2^4 代表十进制数16

0×2^3 代表十进制数0

0×2^2 代表十进制数0

1×2^1 代表十进制数2

1×2^0 代表十进制数1

1×2^{-1} 代表十进制数0.5

0×2^{-2} 代表十进制数0

1×2^{-3} 代表十进制数0.125

（10110011.101）$_2$＝128＋32＋16＋2＋1＋0.5＋0.125

＝（179.625）$_{10}$

非十进制数转换成十进制数

非十进制数转换成十进制数的方法是：把各个非十进制数按权展开求和。

二进制数与八进制数之间的转换

二进制数与八进制数之间的转换十分简捷方便，他们之间的对应关系是：八进制数的每一位对应二进制数的三位，$8=2^3$。

二进制数转换成八进制数

将二进制数从小数点开始，整数部分从右向左3位一组，小数部分从左向右3位一组，不足三位用0补足。

将（10110101110.11011）$_2$转换为八进制数

010	110	101	110	.	110	110
↓	↓	↓	↓		↓	↓
2	6	5	6	.	6	6

（10110101110.11011）$_2$＝（2656.66）$_8$

八进制数转换成二进制数

以小数点为界，向左或向右每一位八进制数用相应的三位二进制数替代，然后将其连在一起。

将（6237.431）$_8$转换为二进制数

6	2	3	7	.	4	3	1
↓	↓	↓	↓		↓	↓	↓
110	010	011	111	.	100	011	001

（6237.431）$_8$＝（110010011111.100011001）$_2$

二进制数与十六进制数之间的转换

二进制数与十六进制数之间的转换十分简捷方便，他们之间的对应关系是：十六进制数的每一位对应二进制数的四位，$16=2^4$。

二进制数转换成十六进制数

将二进制数从小数点开始，整数部分从右向左4位一组，小数部分从左向右4位一组，不足四位用0补足。

将（101001010111.110110101）$_2$转换为十六进制数

1010	0101	0111	.	1101	1010	1000
↓	↓	↓		↓	↓	↓
A	5	7	.	D	A	8

（101001010111）$_2$＝（A57.DA1）$_{16}$

十六进制数转换成二进制数

以小数点为界，向左或向右每一位十六进制数用相应的四位二进制数替代，然后将其连在一起即可。

将（3AB.11）$_{16}$转换成二进制数

3	A	B	.	1	1
↓	↓	↓		↓	↓
0011	1010	1011	.	0001	0001

（3AB.11）$_{16}$＝（1110101011.00010001）$_2$

二进制数的算术运算

二进制数的算术运算（一位二进制数）

- 加：　0 + 0=0　0 + 1=1　1 + 0=1　1 + 1=0
- 减：　0 - 0=0　1 - 1=0　1 - 0=1　0 - 1=1
- 乘：　0×0=0　0×1=0　1×0=0　1×1=1
- 除：　0÷0　0÷1=0　1÷0　1÷1=1

例：

111 + 1010=(?)　　10001

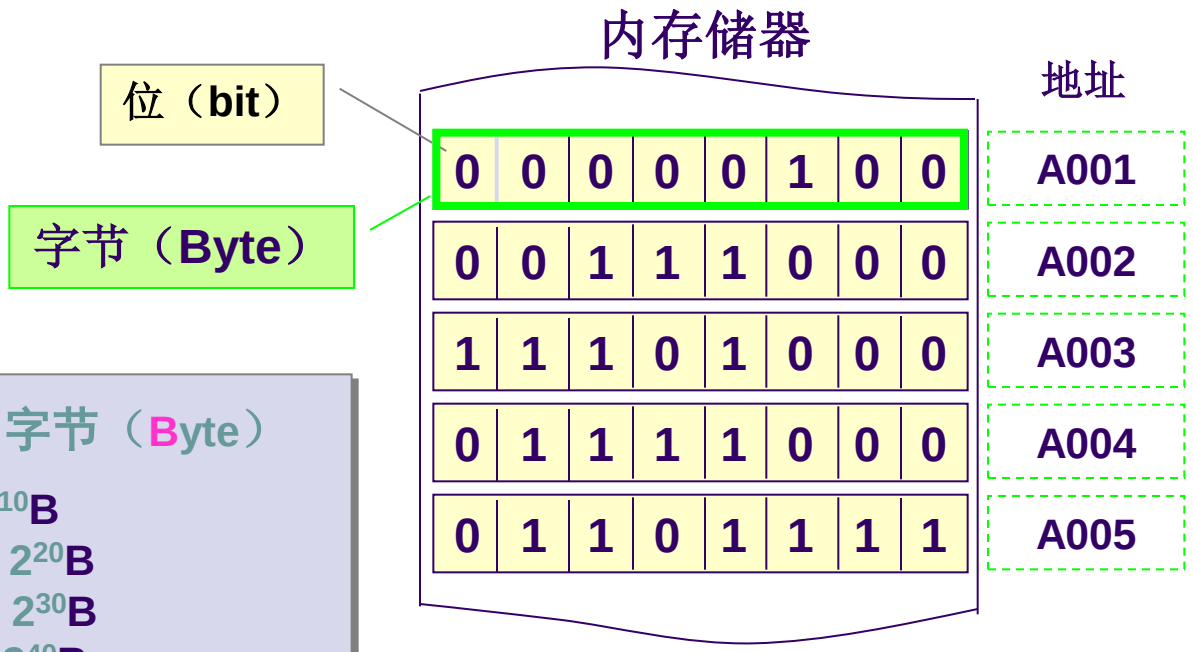
10.1×1010=(?)　　11001

不同进制数之间的转换和计算

一个实用的工具：Windows操作系统的计算器程序

数据在计算机中的表示

在计算机内部，采用二进制的形式表示数据，用具有固定长度的单元存储数据。



基本存储单位：字节（Byte）

1KB = 1024B = 2^{10}B
1MB = 1024KB = 2^{20}B
1GB = 1024MB = 2^{30}B
1TB = 1024GB = 2^{40}B
1PB = 1024TB = 2^{50}B

整数的表示

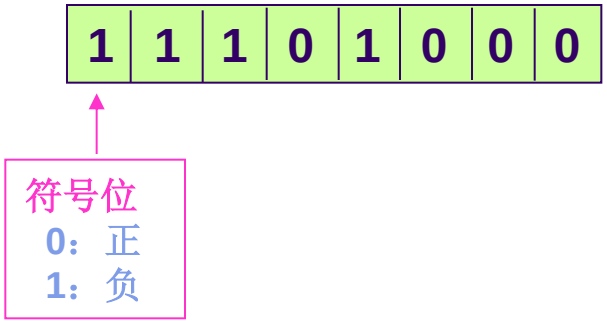
一般用定点数表示

定点数指小数点在数中有固定的位置。

整数可分为无符号整数和(有符号)整数

无符号整数（不带符号的整数）中，所有二进制位用来表示数的大小。

整数（带符号的整数）用最高位表示数的正负号，其它位表示数的大小。



无符号数：（11101000）$_2$

有符号数：（-1101000）$_2$

整数的取值范围

◆如果用一个字节表示一个无符号整数，其取值范围是0 ~ 255（2^8-1）。

（00000000）$_2$ ~（11111111）$_2$

◆若表示一个有符号整数，其取值范围是-128 ~ +127（-2^7 ~ 2^7-1）。

（10000000）$_2$ ~（01111111）$_2$

※如果某个数的数值超出了某一个数值范围就称为“溢出”。

如果用一个字节表示整数，则能表示的最大正整数为01111111（最高位为符号位），即最大值为127，若数值＞127，则“溢出”。

例如：128就溢出了。

整数的原码表示

◆四位整数的原码

十进制表示	原码表示	十进制表示	原码表示
+8	-	-8	-
+7	0111	-7	1111
+6	0110	-6	1110
+5	0101	-5	1101
+4	0100	-4	1100
+3	0011	-3	1011
+2	0010	-2	1010
+1	0001	-1	1001
+0	0000	-0	1000

$$A = \begin{cases} \sum_{i=1}^{n-1} 2^{i-1} a_i & \text{若} a_n=0 \\ -\sum_{i=1}^{n-1} 2^{i-1} a_i & \text{若} a_n=1 \end{cases}$$

缺点：

加减运算时须考虑符号。

有两个零。

整数的补码表示

◆四位整数的补码

十进制表示	补码表示	十进制表示	补码表示
+8	-	-8	1000
+7	0111	-7	1001
+6	0110	-6	1010
+5	0101	-5	1011
+4	0100	-4	1100
+3	0011	-3	1101
+2	0010	-2	1110
+1	0001	-1	1111
+0	0000	-0	-

$$A = -2^{n-1}a_n + \sum_{i=1}^{n-1} 2^{i-1}a_i$$

补码表示时，整数的取负规则：

将整数的每位（包括符号位）求反。

将结果作为一个无符号数对待，加1。

实数的表示

一般用浮点数表示。

因为它的小数点位置不固定，所以称为浮点数。

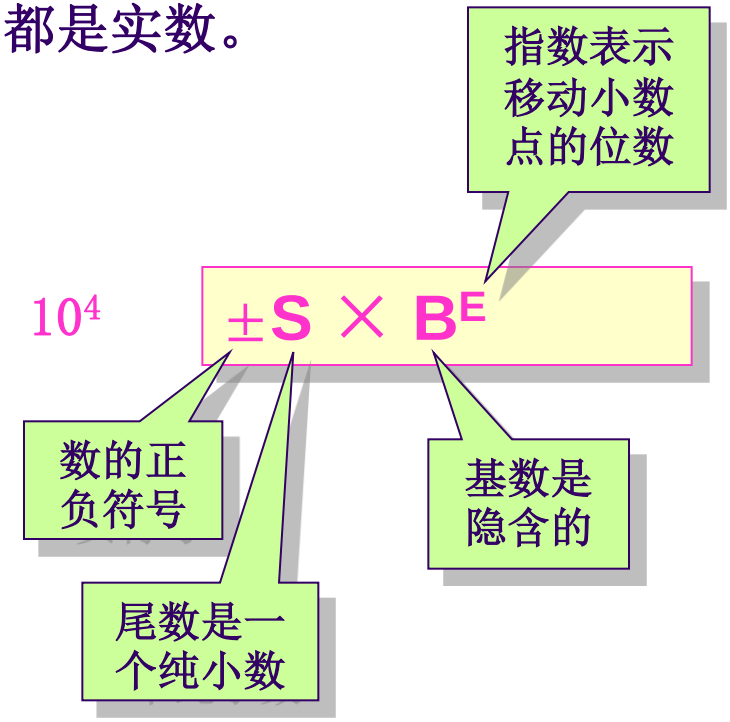
57.625、－1984.045、0.00456都是实数。

以上三个数又可以表示为：

$57.625 = (0.57625) \times 10^{2}$

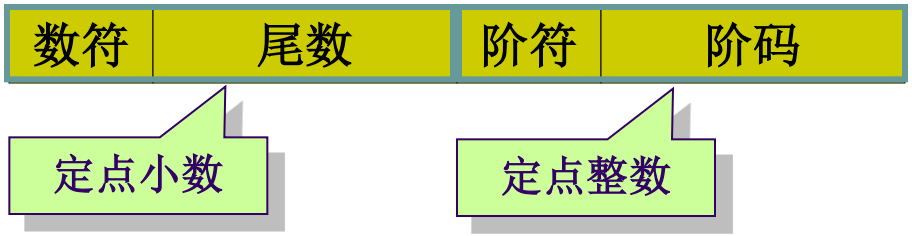
$-1984.045 = (-0.1984045) \times 10^{4}$

$0.00456 = (0.456) \times 10^{-2}$



浮点数的机内表示形式

在计算机中一个浮点数由尾数和指数两部分组成。



尾数表示数值的有效数字，小数点固定在数符和尾数之间；

阶码用来指示尾数中的小数点应当向左或向右移动的位数；

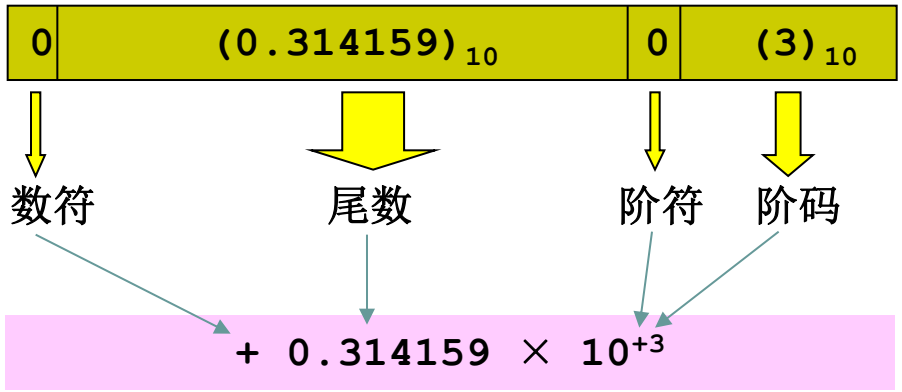
在浮点数中数符和阶符各占一位。

尾数部分的宽度决定了有效数字的位数（精度），

阶码部分的宽度决定了数值范围。

浮点数的机内表示形式

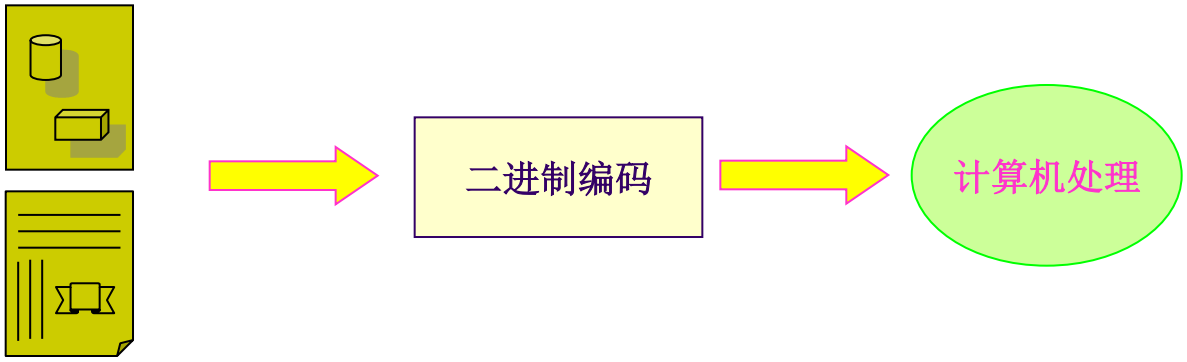
举例：314.159



字符在计算机中的表示

字符编码（Character Code）是用二进制编码来表示字母、数字以及专门符号。

编码是采用少量的基本符号，选用一定的组合原则，以表示大量复杂多样的信息的技术。



ASCII码

常用的西文编码是ASCII（American Standard Code for Information Interchange）码，即美国信息交换标准代码。

ASCII码有7位版本和8位版本两种，通用的是7位版本。

7位版本的ASCII码有128个元素，只需用7个二进制位（2^7＝128）表示，其中控制字符32个，阿拉伯数字10个，大小写英文字母52个，各种

码、数与数值

在计算机中实际用8位表示一个字符，最高位为“0”。

例如：数字0的ASCII码为48，大写英文字母A的ASCII码为65，空格的ASCII码为32等等。有的计算机教材中的ASCII码用16进制数表示，这样，数字0的ASCII码为30H，字母A的ASCII为41H，…。

汉字编码

汉字也是字符，与西文字符比较，汉字数量大，字形复杂，同音字多，这就给汉字在计算机内部的存储、汉字的传输与交换、汉字的输入、输出等带来了一系列的问题。

为了能直接使用西文标准键盘输入汉字，必须为汉字设计相应的编码，以适应计算机处理汉字的需要。

汉字的处理过程

汉字的处理过程如下图所示：

输入码

- 音码：全拼、双拼、微软拼音、智能ABC等。
- 形码：五笔字型、郑码等。
- ……

国标码

1980年我国颁布了《信息交换用汉字编码字符集 基本集》代号为（GB2312－80），是国家规定的用于汉字信息处理使用的代码依据，这种编码称为国标码。

在国标码的字符集中共收录了6763个常用汉字和682个非汉字字符（图形、符号），其中一级汉字3755个，以汉语拼音为序排列，二级汉字3008个，以偏旁部首进行排列。

区位码

国标GB2312－80规定，所有的国标汉字与符号组成一个94×94的矩阵，在此方阵中，每一行称为一个“区”（区号为01－94），每一列称为一个“位”（位号为01－94），该方阵实际组成了一个94个区，每个区内有94个位的汉字字符集，每一个汉字或符号在码表中都有一个唯一的位置编码，叫该字符的区位码。

使用区位码方法输入汉字时，必须先在表中查找汉字并找出对应的代码，才能输入。区位码输入汉字的优点是无重码，而且输入码与内部编码的转换方便。

区 \ 位	01	02	…	94
01				
02				
…	啊(1601)	阿(1602)		
94				

国标码与区位码

国标码并不等于区位码，它是由区位码稍作转换得到,其转换方法为：

先将十进制区码和位码转换为十六进制的区码和位码，这样就得了一个与国标码有一个相对位置差的代码；

再将这个代码的第一个字节和第二个字节分别加上**20H**，就得到国标码。

举例：“保” 字的国标码为**3123**H，它是经过下面的转换得到的：

17 03 D	区位码（十进制）
11 03 H	区位码（十六进制）
+20 20 H	加一个常量
31 23 H	国标码

机内码

汉字的机内码是计算机系统内部对汉字进行存储、处理、传输统一使用的代码，又称为汉字内码。

用2个字节来存放汉字的内码。在计算机内汉字字符必须与英文字符区别开，以免造成混乱。英文字符的机内码是用一个字节来存放ASCII码，一个ASCII码占一个字节的低7位，最高位为“0” 。汉字机内码中两个字节的最高位均置“1”。

例如：汉字“保”

国标码为3123H （00110001 00100011）$_2$

\+ 8080H （10000000 10000000）$_2$

机内码为B1A3H （10110001 10100011）$_2$

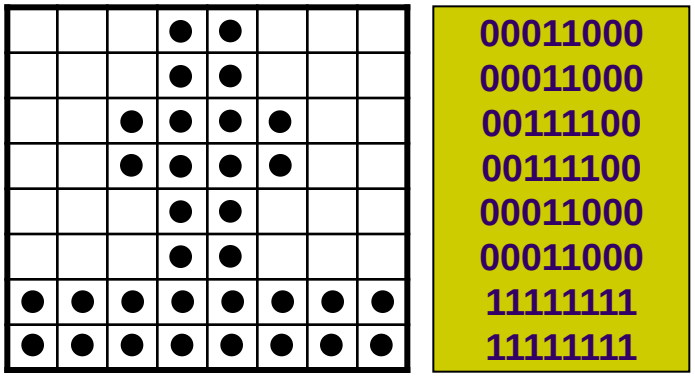
字形码

每一个汉字的字形都必须预先存放在计算机内，例如GB2312国标汉字字符集的所有字符的形状描述信息集合在一起，称为字形信息库，简称字库。

目前汉字字形的产生方式大多是用点阵方式形成汉字。

根据汉字输出精度的要求，有不同密度点阵。例如16×16点阵、24×24点阵等。

汉字字形点阵中每个点的信息用一位二进制码来表示，“1”表示对应位置处是黑点，“0”表示对应位置处是空白。



字形码

地址码

字形点阵的信息量很大，所占存储空间也很大。

16×16点阵，每个汉字就要占32个字节（16×16÷8＝32）

24×24点阵的字形码需要用72字节（24×24÷8＝72）

因此字形点阵只能用来构成“字库”，而不能用来替代机内码用于机内存储。

字库中存储了每个汉字的字形点阵代码，不同的字体（如宋体、仿宋、楷体、黑体等）对应着不同的字库。

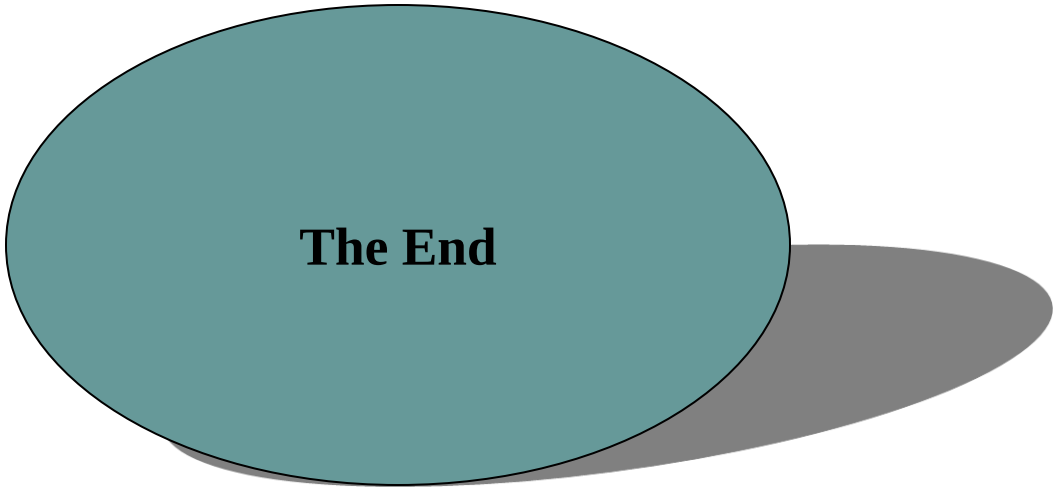
字形码在字库中的相对位置就是地址码。地址码和机内码之间要有简明的对应转换关系。

其它汉字编码

GBK：汉字扩充编码，能支持两万多汉字。

Unicode：针对各国文字、符号进行统一编码。用两字节表示。

BIG5：台湾、香港地区采用的繁体汉字编码。


The End