

湖北省技能高考理论知识

计算机类

第二章

计算机中信息的表示形式

第二章 计算机中信息的表示形式

2.1 信息与数据

2.2 计算机中的进制数

2.3 数值型数据在计算机中的表示形式

2.4 非数值数据在计算机中的表示

2.5 数据存储容量单位

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1 信息与数据

2.1.1 信息与数据的概念

2.1.2 信息的特点

2.1.3 人类社会的五次信息革命

2.1.4 信息技术IT

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

2.1.1 信息与数据的概念

- 信息是客观事物的属性和相互联系特性的表现,反映了客观事物的存在或运动状态。通常是指音讯、消息,泛指人类传播的一切内容,是看不见摸不着的存在于人类大脑中的一切东西。
- 信息奠基人香农(Shannon) (被称为“信息论之父”)认为“信息是用来消除随机不确定性的东西”。
- 控制论创始人维纳 (Norbert Wiener) 认为“信息是人们在适应外部世界,并使这种适应反作用于外部世界的过程中,同外部世界进行互相交换的内容和名称”。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.1 信息与数据的概念

- 数据是反映客观事物属性的记录, 是信息的载体, 信息的具体表现形式。
- 数据有多种表现形式, 如数字、字符、声音、图像等。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.1 信息与数据的概念

- 信息与数据是不可分离的。信息由与物理介质有关的数据表达,数据中所包含的意义就是信息。信息是对数据的解释、运用,是经过处理以后的数据,数据只有经过解释才有意义,才成为信息。
- 数据是原始事实,而信息是数据处理的结果。不同知识、经验的人,对于同一数据的理解,可得到不同信息。数据经加工处理后成为信息,而信息必须通过数据才能传播,才能对人类起作用。信息是有意义的,数据是无意义的。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.2 信息的特点

- 客观性:信息是事物的特征和变化的客观反映。
- 时效性:信息时过境迁往往就失去价值。
- 普遍性:信息无所不在, 无处不有。
- 另外信息还具有: 可传递性、依附性、可处理性、真伪性、共享性、价值相对性等特征。

[目录](#)

[上一页](#)

[下一页](#)

[结束](#)

2.1.3 人类社会的五次信息革命

- 第一次：“语言”的产生，是从猿进化到人的重要标志；自从有了人类，就产生了信息处理技术。
- 第二次：“文字”的产生，使信息的存储和传递首次超越了时间和地域的局限。
- 第三次：“纸”和“印刷术”的发明，为知识的积累和传播提供了更为可靠的保证。
- 第四次：电话电报电视等通信技术的发明与应用，进一步突破了时间和空间的限制。
- 第五次：“信息化技术”的使用，将人类社会推进到了数字化的信息时代。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.4 信息技术IT

- 信息技术是指有关信息的收集、识别、提取、变换、存储、处理、检索、检测、分析和利用等的技术，主要包括传感技术、计算机技术、控制技术和通信技术，称为现代信息技术四基元。
- 3C技术:通信技术、计算机技术和控制技术 (Communication, Computer, Control)的合称, 3C技术是信息技术的主体。
- 微电子技术是现代信息技术的基石。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.4 信息技术IT

- 信息产业，以计算机和通讯设备行业为主体的IT产业, 我们通常称之为信息产业, 又称为第四产业。
- 信息社会的三大资源：物质、能源和信息。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.1.4 信息技术IT

- 媒体是指人借助用来传递信息与获取信息的工具、渠道、载体、中介物或技术手段。
- 第四媒体，是指网络，是继报纸、广播、电视多种传统大众传播媒体之后出现的一种新的大众传播媒体。
- 自媒体(外文名：We Media)又称“公民媒体”或“个人媒体”，是指私人化、平民化、普泛化、自主化的传播者，以现代化、电子化的手段，向不特定的大多数或者特定的单个人传递规范性及非规范性信息的新媒体的总称。自媒体平台包括：博客、微博、微信、抖音、百度官方贴吧、论坛/BBS等网络社区。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2 进制数的基本概念

2.2.1 进位计数制

2.2.2 进制数转换

2.2.3 二进制数的算术运算

2.2.4 二进制数的逻辑运算

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.1 数制及其转换

进位计数制也称计数制，是指用一组固定的符号和统一的规则来表示数值的方法。计算机是信息处理的工具，任何信息必须转换成二进制形式表示的数据后才能由计算机进行处理、存储和传输。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

数制中的几个名词术语

- 数码：一组用来表示某种数制的符号。如：1、2、3、4、A、B、C、I、II、III、IV等。
- 基数：数制所使用的数码个数称为“基数”或“基”，常用“R”表示，称为R进制。如二进制的数码是0、1，基为2。
- 位权：一种进制数中每位所代表的实际大小称为位权，一般用 R^i 来表示。
- 数位是指某位在数中的位置，一般用 R_i 来表示。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

计算机中采用二进制形式表示数据的原因

- 在物理上最容易实现。这是计算机中采用二进制数来表示信息的主要原因。
- 用来表示的二进制数的编码、计数、加减运算规则简单。
- 逻辑判断准确。
- 机器可靠性高。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.2 数制及其转换

常见的几种进位计数制

1. 十进制 (**D**ecimal System) : 由0、1、2、...、8、9十个数码组成, 即基数为10。特点为: 逢十进一, 借一当十。用字母D表示。

2. 二进制 (**B**inary System) : 由0、1两个数码组成, 即基数为2。二进制的特点为: 逢二进一, 借一当二。用字母B表示。

3. 八进制 (**O**ctal System) : 由0、1、2、3、4、5、6、7八个数码组成, 即基数为8。八进制的特点为: 逢八进一, 借一当八。用字母O表示。

4. 十六进制 (**H**exadecimal System) : 由0、1、2、...、9、A、B、C、D、E、F十六个数码组成, 即基数为16。十六进制的特点为: 逢十六进一, 借一当十六。用字母H表示。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

各种进制之间的对应关系见表2-1

表2-1 十进制、二进制、八进制、十六进制之间的对应关系

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	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8	17	10001	21	11

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

2.2.2 数制及其转换

数制的转换

1. 二进制、八进制、十六进制数转化为十进制数

对于任何一个二进制数、八进制数、十六进制数，均可以先写出它的位权展开式，然后再按十进制进行计算即可将其转换为十进制数。

例如：

$$(1111.11)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 15.75$$

$$(A10B.8)_{16} = 10 \times 16^3 + 1 \times 16^2 + 0 \times 16^1 + 11 \times 16^0 + 8 \times 16^{-1} = 41\,227.5$$

注意：在不至于产生歧义时，可以不注明十进制数的进制，如上例。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

2.2.2 数制及其转换

2. 十进制数转化为二进制数

十进制数的整数部分和小数部分在转换时需作不同的计算，分别求值后再组合。

整数部分采用除2取余法，即逐次除以2，直至商为0，得出的余数倒排，即为二进制各位的数码。小数部分采用乘2取整法，即逐次乘以2，从每次乘积的整数部分得到二进制数各位的数码。(参见下例)

例：将十进制数100.125转化为二进制数

[目 录](#)

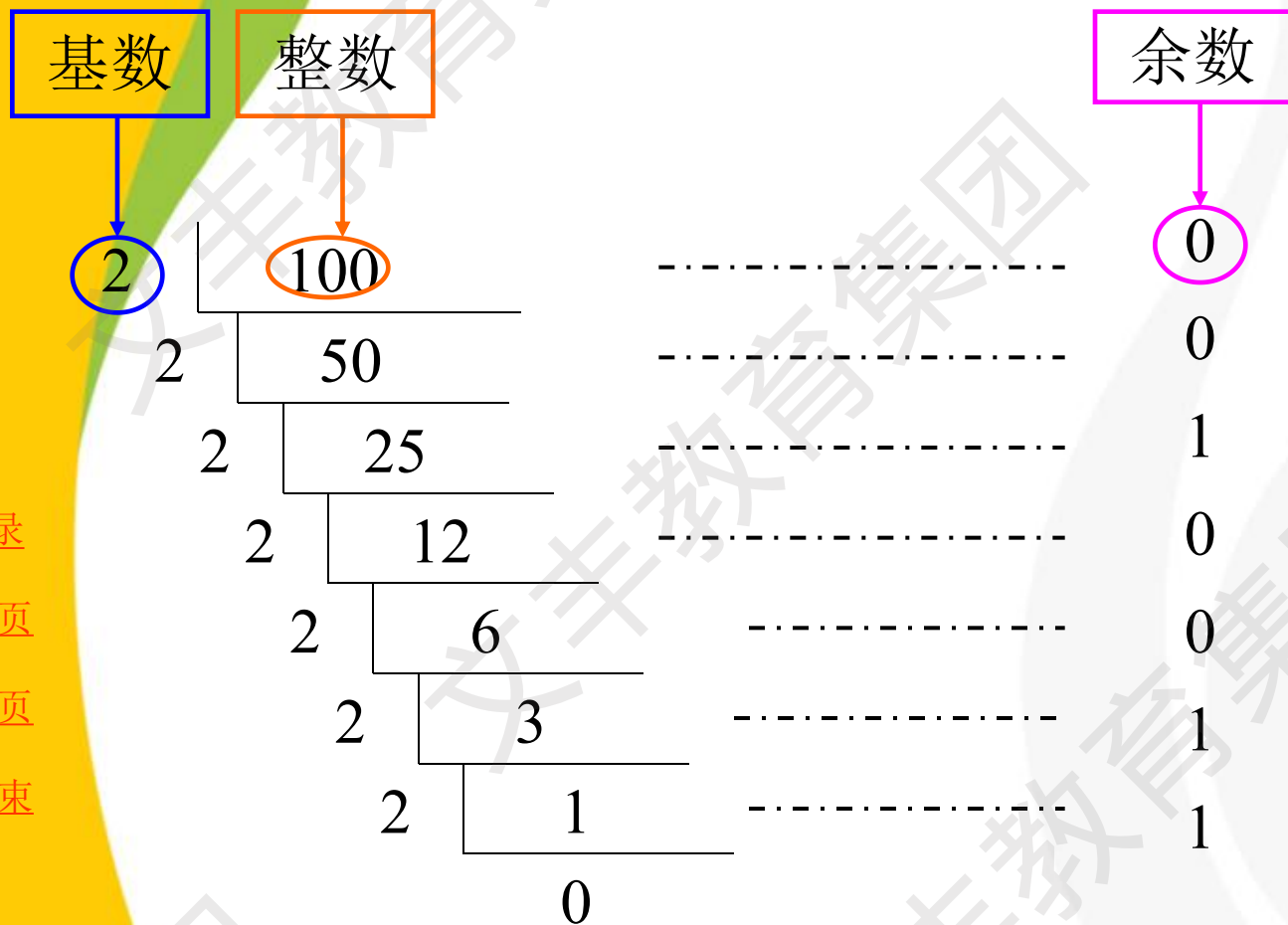
[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

步骤一：先对整数100进行转换：



[目录](#)

[上一页](#)

[下一页](#)

[结束](#)

由上得出， **100D = 1100100B**

步骤二：对于小数部分0.125的转换

$$0.125 \times 2 = 0.250$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

由上得出， $0.125D=0.001B$ 。

$0 \dots a_{-1}$

$0 \dots a_{-2}$

$1 \dots a_{-3}$



将整数和小数部分组合，得出： $100.125D = 1100100.001B$ 。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.2 数制及其转换

3. 二进制数与八进制数的相互转换

二进制数转换成八进制数的方法是：将二进制数从小数点开始，对二进制整数部分向左每3位分成一组，不足3位的向高位补0凑成3位；对二进制小数部分向右每3位分成一组，不足3位的向低位补0凑成3位。每一组有3位二进制数，分别转换成八进制数码中的一个数字，全部连接起来即可。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.2 数制及其转换

例：把二进制数1111101.101转化为八进制数。

二进制3位分组	011	111	101.	101
转换为八进制数	3	7	5.	5

所以，**1111101.101B = 375.5O。**

将八进制数转换成二进制数，只要将每一位八进制数转换成相应的3位二进制数，依次连接起来即可。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.2 数制及其转换

4. 二进制数与十六进制数的相互转换

二进制数转换成十六进制数，只要把每4位分成一组，再分别转换成十六进制数码中的一个数字，不足4位的分别向高位或低位补0凑成4位，全部连接起来即可。

十六进制数转换成二进制数，只要将每一位十六进制数转换成4位二进制数，然后依次连接起来即可。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.2 数制及其转换

例：将10110001.101B转换为十六进制数。

二进制4位分组	1011	0001.	1010
转换为十六进制数	B	1.	A

所以，**10110001.101 B = B1.AH。**

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

进制数之间转换的结论

(1) 记忆权值的变化规律:

R进制数，整数部分第n位的权值是 R^{n-1} ，小数部分第n位的权值是 R^{-n} 。

(2) 不同进制数之间转换后，数据位数的变化情况

十进制数转换成R进制数后，数位会随着R变大而减少。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

进制数之间转换的结论

(3) 有限数位的问题

十进制有限数位的小数转换成二进制小数不一定就是有限数位，但是二进制（包括 2^n 进制）的有限数位的小数转换成十进制小数一定是有限数位的。

(4) 分数的表示方法

计算机中分数不能直接表示出来，要先转换成小数。例如：

$$(11/64)_{10} = (0.001011)_2$$

(5) 一个 n 位无符号二进制数，其中各位上都是“1”，转换成十进制数就是 $2^n - 1$ 。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.3 数制及其转换

二进制的运算规则

1. 算术运算规则

加法规则： $0 + 0 = 0$ ； $0 + 1 = 1$ ；
 $1 + 0 = 1$ ； $1 + 1 = 10$ （向高位有进位）；

减法规则： $0 - 0 = 0$ ； $10 - 1 = 1$ （向高位借位）；
 $1 - 0 = 1$ ； $1 - 1 = 0$ ；

乘法规则： $0 \times 0 = 0$ ； $0 \times 1 = 0$ ；
 $1 \times 0 = 0$ ； $1 \times 1 = 1$

除法规则： $0 / 1 = 0$ ； $1 / 1 = 1$

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.2.4 数制及其转换

2. 逻辑运算规则

非运算 (NOT) :

与运算 (AND) : $0 \wedge 0 = 0$; $0 \wedge 1 = 0$;
 $1 \wedge 0 = 0$; $1 \wedge 1 = 1$;

或运算 (OR) : $0 \vee 0 = 0$; $0 \vee 1 = 1$;
 $1 \vee 0 = 1$; $1 \vee 1 = 1$;

异或运算 (XOR) : $0 \oplus 0 = 0$; $0 \oplus 1 = 1$;
 $1 \oplus 0 = 1$; $1 \oplus 1 = 0$;

同或运算 : $0 \odot 0 = 1$; $0 \odot 1 = 0$;
 $1 \odot 0 = 0$; $1 \odot 1 = 1$;

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

逻辑运算

表2-2 逻辑运算真值表

a	b	$a \bullet b$	$a + b$	$a \oplus b$	$a \odot b$
0	0	0	0	0	1
1	1	1	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3 数值型数据在计算机中的表示形式

2.3.1 整数在计算机中的表示形式

2.3.2 计算机中溢出的概念

2.3.3 计算机中实数的表示形式

2.3.4 二进制数的编码

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

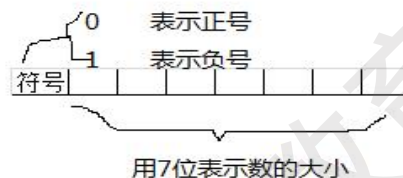
2.3.1 整数在计算机中的表示形式

真值

一般书写中用“+”、“-”来表示数的符号，这样表示的数称为真值。

机器数

用二进制数“0”或“1”来表示数的符号，“0”表示正号，“1”表示负号，且把符号位置于该数的最高数值位之前，这样把符号位和数值位一起编码来表示的数就是机器数（或称为机器码）。



[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

原码

符号位为0表示正数，为1表示负数，数值部分用二进制数的绝对值表示的方法称为原码表示法，通常用 $[X]_{\text{原}}$ 表示X的原码。原码表示的特点：

- ①最高位为符号位，正数为0，负数为1；
- ②8位二进制原码表示数的范围是 $-127 \sim +127$ ，十六位二进制原码表示数的范围是 $-32767 \sim +32767$ ；
- ③0的原码有两种表示方法，即+0和-0，设字长为8位：

$[+0]_{\text{原}} = 00000000\text{B}$

$[-0]_{\text{原}} = 10000000\text{B}$

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

反码

正数的反码与原码相同，负数的反码是符号位不变，数值位逐位取反。反码表示的特点：

- ①反码表示法中，最高位仍为符号位，正数为0，负数为1；
- ②“0”有两种表示方法：当字长是8位时
 $[+0]_{\text{反}} = 00000000\text{B}$ ， $[-0]_{\text{反}} = 11111111\text{B}$
- ③8位二进制反码表示数的范围是 $-127 \sim +127$ 。
- ④正数的反码与原码相同，负数的反码符号位为1，其数值部分按位取反。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

补码

正数的补码与其原码相同；负数的补码是符号位不变，数值位逐位取反（即求其反码），然后在最低位加1。补码表示的特点：

- ①补码表示中，最高位仍为符号位，正数为0，负数为1；
- ②0仅有一种表示方法，即 $[+0]_{\text{补}} = [-0]_{\text{补}}$ ；
- ③8位二进制补码表示数的范围是 $-128 \sim +127$ ，十六位二进制补码表示数的范围是 $-32768 \sim +32767$ ；对于同一个数，作为8位二进制数的补码和作为16位二进制数的补码不同，这一点要特别注意。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

现代计算机中采用二进制数补码形式表示的原因：

- 使用补码，可以将符号位和数值域统一处理；
- 同时，加法和减法也可以统一处理，于是可以简化运算单元的设计。
- 此外，补码与原码相互转换，其运算过程是相同的，不需要额外的硬件电路。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

※移码

移码（又叫增码）是符号位取反的补码，一般用做浮点数的阶码，引入的目的是为了保证浮点数的机器零为全0。等于补码的符号（第一位数字）位取反。先把这个数化为补码，然后再根据补码的第一位数字取反即可。移码和补码的关系：同一数值的移码与补码符号位相反，其它各位相同。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

总结:

- 正数的原码，补码，反码相同；
- 负数的反码：原码的数值取反；
- 负数的补码：原码转换成反码，反码末位加1
- 数的移码：与补码的符号位（第一位数字）取反

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.1 整数在计算机中的表示形式

表2-3 计算机中整数的表示范围

位 数	无符号数	机 器 数		
		原码	反码	补码
1个字节	0—255	-127—127	-127—127	-128—127
2个字节	0—65535	-32767—32767	-32767—32767	-32768—32767
4个字节	0— $2^{32}-1$	$-(2^{31}-1) \text{—} 2^{31}-1$	$-(2^{31}-1) \text{—} 2^{31}-1$	$-2^{31} \text{—} 2^{31}-1$
N位	$0 \text{—} 2^n-1$	$-(2^{n-1}-1) \text{—} 2^{n-1}-1$	$-(2^{n-1}-1) \text{—} 2^{n-1}-1$	$-2^{n-1} \text{—} 2^{n-1}-1$

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.2 计算机中溢出的概念

- 溢出，是指带符号数的加、减运算的结果超出了字长所能表示的范围。
- 若发生了溢出，则运算结果必然是错误的，正常情况下溢出是不允许的，必须要进行修正。
- 两个正数相加，结果大于机器所能表示的最大正数，称为正溢出（又称上溢）；而两个负数相加，结果小于机器所能表示的最小负数，称为负溢出（又称下溢）。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.3 计算机中实数的表示形式

定点数和浮点数

- 计算机中常用的数值数据表示格式有两种，定点格式和浮点格式。
- 所谓定点数和浮点数，是指在计算机中一个数的小数点的位置是固定的还是浮动的：如果一个数中小数点的位置是固定的，则为定点数；如果一个数中小数点的位置是浮动的，则为浮点数。
- 一般来说，定点格式可表示的数值的范围有限，但要求的处理硬件比较简单。而浮点格式可表示的数值的范围很大，但要求的处理硬件比较复杂。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.3 计算机中实数的表示形式

浮点数的格式包括：阶码和尾数

尾符	尾数	阶符	阶码
----	----	----	----

- 在计算机中表示一个浮点数时，一是要给出尾数M，用纯小数形式表示；二是要给出指数e，用整数形式表示，常称为阶码。
- 尾数部分给出有效数字的位数，因而决定了浮点数的表示精度；阶码部分指明了小数点在数据中的位置，因而决定了浮点数的表示范围。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.4 二进制数的编码

BCD编码

- 亦称8421BCD码，是用4位二进制数来表示1位十进制数中的0~9这10个数码的一种表示方法。
- 是一种二进制的数字编码形式，用二进制形式编码的十进制代码。

表2-4 BCD码

十进制	BCD (8421) 码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.3.4 二进制数的编码

奇偶校验码

- 奇偶校验是一种校验代码传输正确性的方法。根据被传输的一组二进制代码的数位中“1”的个数是奇数或偶数来进行校验。采用奇数的称为奇校验，反之，称为偶校验。
- 奇偶校验能够检测出信息传输过程中的部分误码（奇数位误码能检出，偶数位误码不能检出，即奇偶校验码只能发现奇数位错误），且不能进行纠错。

[目录](#)

[上一页](#)

[下一页](#)

[结束](#)

2.4 非数值型数据在计算机中的表示

2.4.1 ASCII码

2.4.2 汉字编码

※2.4.3 Unicode编码

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.4.1 ASCII码

- 目前计算机中最广泛使用的字符编码是ASCII码，是美国标准信息交换码(American Standard Code for Information Interchange)的英文缩写。ASCII码分为标准ASCII码和扩展ASCII码两种。
- 标准ASCII码使用7个二进制位对字符进行编码，扩展ASCII码使用8位二进制位对字符进行编码。
- 每个ASCII字符以一个字节存放在计算机中，标准ASCII码最高位为0，扩展ASCII码最高位可以为0或1。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.4.1 ASCII码

- 标准ASCII码字符集共有128个字符，其中包含：94个可打印字符（包括常用字母、数字、标点符号等），34个控制字符。
- 一般要记住几个特殊字符的ASCII码（用于表的定位）：空格（32、20H）、A（65、41H）、a（97、61H）、0（48、30H）、（127 7FH Del）。
- ASCII码表中数字、字母的ASCII码值，由小到大依次排列为：空格<数字<大写字母<小写字母；数字、字母的ASCII码是连续排列的，即0—9、a—z、A—Z依次连续排列；对应大小写字母ASCII码值相差32（20H）。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.4.2 汉字编码

- 外码也叫输入码，是用来将汉字输入到计算机中的一组键盘符号，可以是各种形式的编码，如音码、形码、音形码、语音输入、手写输入等。
- 中国标准总局1981年制定了中华人民共和国国家标准GB2312--80《信息交换用汉字编码字符集——基本集》，即国标码。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.4.2 汉字编码

- 区位码是国标码的另一种表现形式，把国标GB2312--80中的汉字、图形符号组成一个 94×94 的方阵，分为94个“区”，每区包含94个“位”，其中“区”的序号由01至94，“位”的序号也是从01至94。每个汉字的区位码是唯一的。
- 在区位码中，01-09区为特殊字符，10-15区为用户自定义符号区（未编码），16-55区为一级汉字，按拼音排序，56-87区为二级汉字，按部首/笔画排序，88-94区为用户自定义汉字区（未编码）。一级常用汉字3755个，按汉语拼音字母顺序排列；二级次常用汉字3008个，按部首排列。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.4.2 汉字编码

- 汉字交换码(国标码)，机内码，汉字内码均用两个字节表示。汉字的区位码、国标码、机内码有如下转换关系：
- 国标码 = 区位码（用16进制数形式表示）+ (2020)H
- 机内码 = 国标码+ (8080)H（相当于将汉字国标码的最高位置为1）
- 机内码 = 区位码+ (A0A0)H，每个汉字的机内码都会大于BOA0H。

[目录](#)

[上一页](#)

[下一页](#)

[结束](#)

2.4.2 汉字编码

- **字型码**是汉字的输出码，输出汉字时都采用点阵形式（现在计算机中都采用矢量图方式）。
- 无论汉字的笔画多少，每个汉字都可以写在同样大小的方块中，通常用 16×16 、 24×24 、 48×48 点阵来显示汉字。
- 汉字存储点阵存储容量的计算公式为 $n \times n/8$ 。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

※2.4.3 Unicode编码

- Unicode编码于1994年正式公布，又叫统一码、万国码（意为所有字符的编码），就是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。
- UTF-8（8位 Unicode）就是在互联网上使用最广的一种Unicode编码的实现方式。UTF-8是一种针对Unicode的可变长度字符编码，用1到6个字节编码Unicode字符。它用于在网页上统一显示中文简体、繁体及其它语言（如英文，日文，韩文）等字符，其中收录了20901个中文字符。

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

2.5 计算机中数据存储容量单位

计算机中数据存储容量的单位

1) 位 (bit)

简记为b，也称为比特，是计算机中表示数据的最小单位。一个二进制位只能表示0或1。

2) 字节 (Byte)

字节来自英文Byte，简记为B。字节是存储信息的基本单位。规定1B=8bit。

$$1 \text{ KB} = 2^{10} \text{ B} = 1024 \text{ B} \quad 1 \text{ MB} = 2^{20} \text{ B} = 1024 \text{ KB}$$

$$1 \text{ GB} = 2^{30} \text{ B} = 1024 \text{ MB} \quad 1 \text{ TB} = 2^{40} \text{ B} = 1024 \text{ GB}$$

3) 字 (Word)

一个字通常由一个字节或若干个字节组成。**字长是计算机一次所能处理的二进制位数长度，字长是衡量计算性能的一个重要指标。**

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)

[返 回](#)

谢谢使用！

[目 录](#)

[上一页](#)

[下一页](#)

[结 束](#)