

计算机中的数据

河北师范大学软件学院





- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储



- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储



- 进制也就是进位制，是人们规定的一种进位方法。
对于任何一种进制—— X 进制，就表示某一位置上的数运算时是逢 X 进一位。

二进制数据



- 十进制计数制（0-9，逢10进1；后缀D，常省略后缀D）

123D, -78D, 3.141D, -1.414

一个十进制位

- 二进制计数制（0, 1；逢2进1；后缀B）

1011B, -110B, 101.11011B, -101.11011B

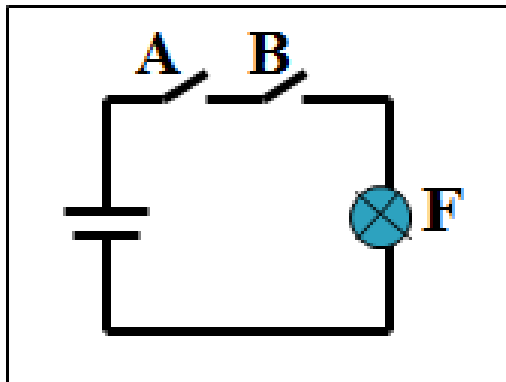
一个二进制位

基本逻辑运算



“与”运算(逻辑乘)
Logic Multiplication

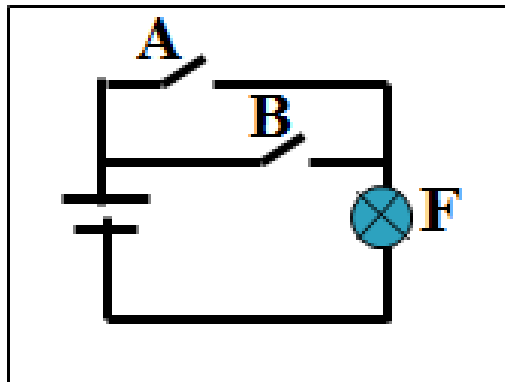
示意
电路



真值
表

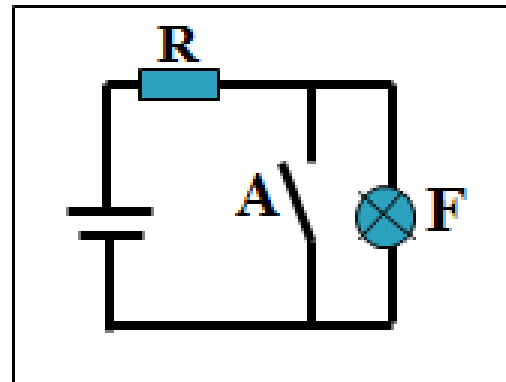
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

“或”运算(逻辑加)
Logic Addition



A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

“非”运算(逻辑非)
Logic Negation



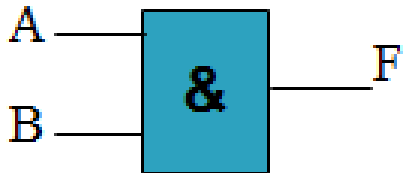
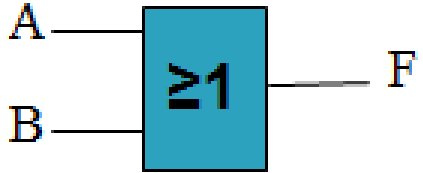
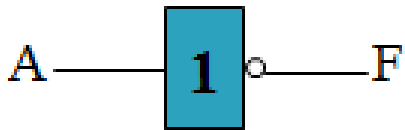
A	F
0	1
1	0



- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储

逻辑电路



	“与”运算(逻辑乘) <i>Logic Multiplication</i>	“或”运算(逻辑加) <i>Logic Addition</i>	“非”运算(逻辑非) <i>Logic Negation</i>
代数式	$F = A \times B = A \cdot B$	$F = A + B$	$F = \overline{A}$
逻辑符号			

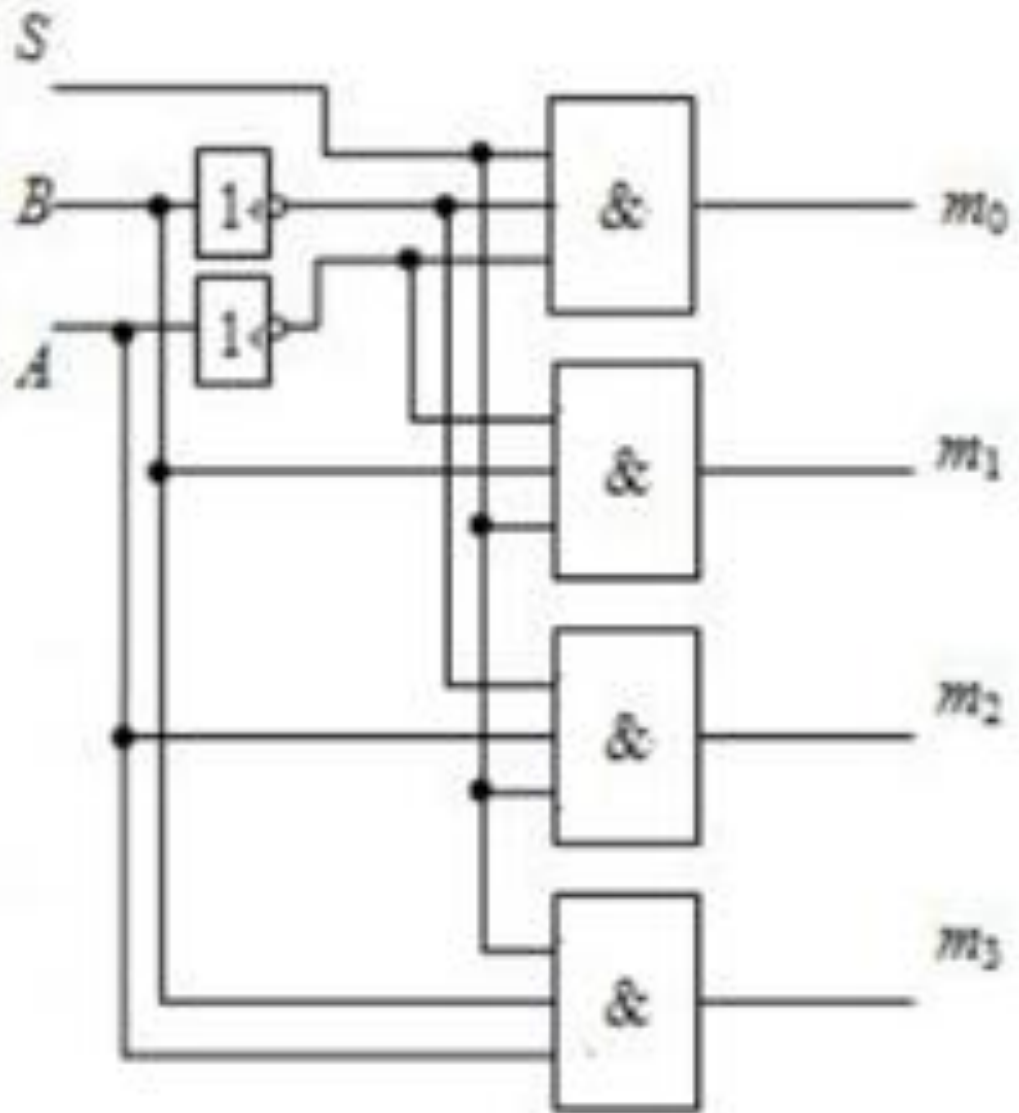


- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储

二四译码器的实现原理



1	B	M_0	7
2			6
	A	M_1	5
		M_2	4
3	S	M_3	





- 容易理解：1根地址线可以产生2 (2^1) 个输出，即可以区分2个内存块，3根地址线可以产生8 (2^3) 个输出，即可以区分8个内存块，4根地址线可以产生16 (2^4) 个输出，即可以区分16个内存块，.....
- 一般的，n根地址线可以产生 2^n 个输出，即可以区分 2^n 个内存块。
- 计算机换算规则：1K= 2^{10} ，1M= 2^{10} K，1G= 2^{10} M



- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储

八进制数据



- 八进制计数制 (0-7; 逢8进1; 后缀Q、0)

-123Q, 777Q, 123.456Q, -61.456Q



一个八进制位

十六进制数据



- 十六进制计数制 (0-9, A(a), B(b), C(c), D(d), E(e), F(f); 逢16进1; 后缀H)

123H, AFH, BC9.4DH, -AD.CH



一个十六进制位



- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储



- R进制转换成十进制
 - 按权展开求和

R进制数M的整数部分有n位，小数部分有m位，转换成十进制数D。

$$\begin{aligned} D &= \pm \sum_{i=-m}^{n-1} (M_i \times R^i) \\ &= \pm (M_{-m} \times R^{-m} + \cdots + M_{n-1} \times R^{n-1}) \end{aligned}$$

其中 M_i 代表第i位的数字，R为**基数**， R^i 为第i位的**权**。



- 二进制转换成十进制

$$\begin{aligned}(1101.11)_2 &= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 1 + 0 + 4 + 8 + 0.5 + 0.25 \\ &= 13.75\end{aligned}$$

- 八进制转换成十进制

$$\begin{aligned}(173.3)_8 &= 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 + 3 \times 8^{-1} \\ &= 64 + 56 + 3 + 0.375 \\ &= 123.375\end{aligned}$$



- 十六进制转换成十进制

$$\begin{aligned}(4C.6)_{16} &= 4 \times 16^1 + 12 \times 16^0 + 6 \times 16^{-1} \\ &= 64 + 12 + 0.375 \\ &= 76.375\end{aligned}$$

- 练习

$$10111.101B = 23.625D$$

$$7035Q = 3613D$$

$$-8FD.CH = -2301.75D$$

数制转换



- 十进制转换成R进制

- 对于整数，通常采用“**除R取余**”的方法，即用R不断地去除要转换的整数，直至**商等于0**为止，然后将所得余数**从后向前**顺序写出，即为转换成的R进制数。

$$\begin{array}{r} 2 \overline{) 43} \\ 2 \overline{) 21} \quad 1 \\ 2 \overline{) 10} \quad 1 \\ 2 \overline{) 5} \quad 0 \\ 2 \overline{) 2} \quad 1 \\ 2 \overline{) 1} \quad 0 \\ 0 \quad 1 \end{array}$$

(余数)

43D = 101011B

$$\begin{array}{r} 8 \overline{) 43} \\ 8 \overline{) 53} \quad 3 \\ 0 \quad 5 \end{array}$$

(余数)

= 53Q

$$\begin{array}{r} 16 \overline{) 43} \\ 16 \overline{) 2B} \\ 0 \quad 2 \end{array}$$

(余数)

= 2BH



• 十进制转换成R进制

$$\begin{array}{rcl}
 - & : & 0.8125 \\
 & * & 2 \\
 \hline
 & : & 1.6250 \text{ 整数部分为1 (最高位)} \\
 & : & 0.625 \\
 & * & 2 \\
 \hline
 & : & 1.250 \text{ 整数部分为1} \\
 & : & 0.25 \\
 & * & 2 \\
 \hline
 & : & 0.50 \text{ 整数部分为0} \\
 & : & 0.5 \\
 & * & 2 \\
 \hline
 & : & 1.0 \text{ 整数部分为1 (最低位)}
 \end{array}$$

的方法，即用 R 不断
分等于0或满足所要
的整数部分，从前
列数。

$$\begin{aligned}
 & \mathbf{0.8125D} \\
 & \mathbf{=0.1101B}
 \end{aligned}$$



- 二进制数据转换成八进制数据
 - 将二进制数据从小数点开始，分别向前向后3位分成一组，不足3位补0，然后写出对应的八进制即可。

$$(101011.101)_2 = \frac{101}{5} \frac{011}{3} \cdot \frac{101}{5} = (53.5)_8$$

- 八进制数据转换成二进制数据
 - 将每位八进制数据写出对应的3位二进制数即可。

$$(173.3)_8 = \frac{1}{001} \frac{7}{111} \frac{3}{011} \cdot \frac{3}{011} = (1111011.011)_2$$



- 二进制数据转换成十六进制数据
 - 将二进制数据从小数点开始，分别向前向后4位分成一组，不足4位补0，然后写出对应的十六进制即可。

$$(101011.101)_2 = \frac{0010}{2} \frac{1011}{B} . \frac{1010}{A} = (2B.A)_{16}$$

- 十六进制数据转换成二进制数据
 - 将每位十六进制数据写出对应的4位二进制数即可。

$$(173.3)H = (\underline{1} \quad \underline{7} \quad \underline{3} . \underline{3}) = (000101110011.0011)B$$

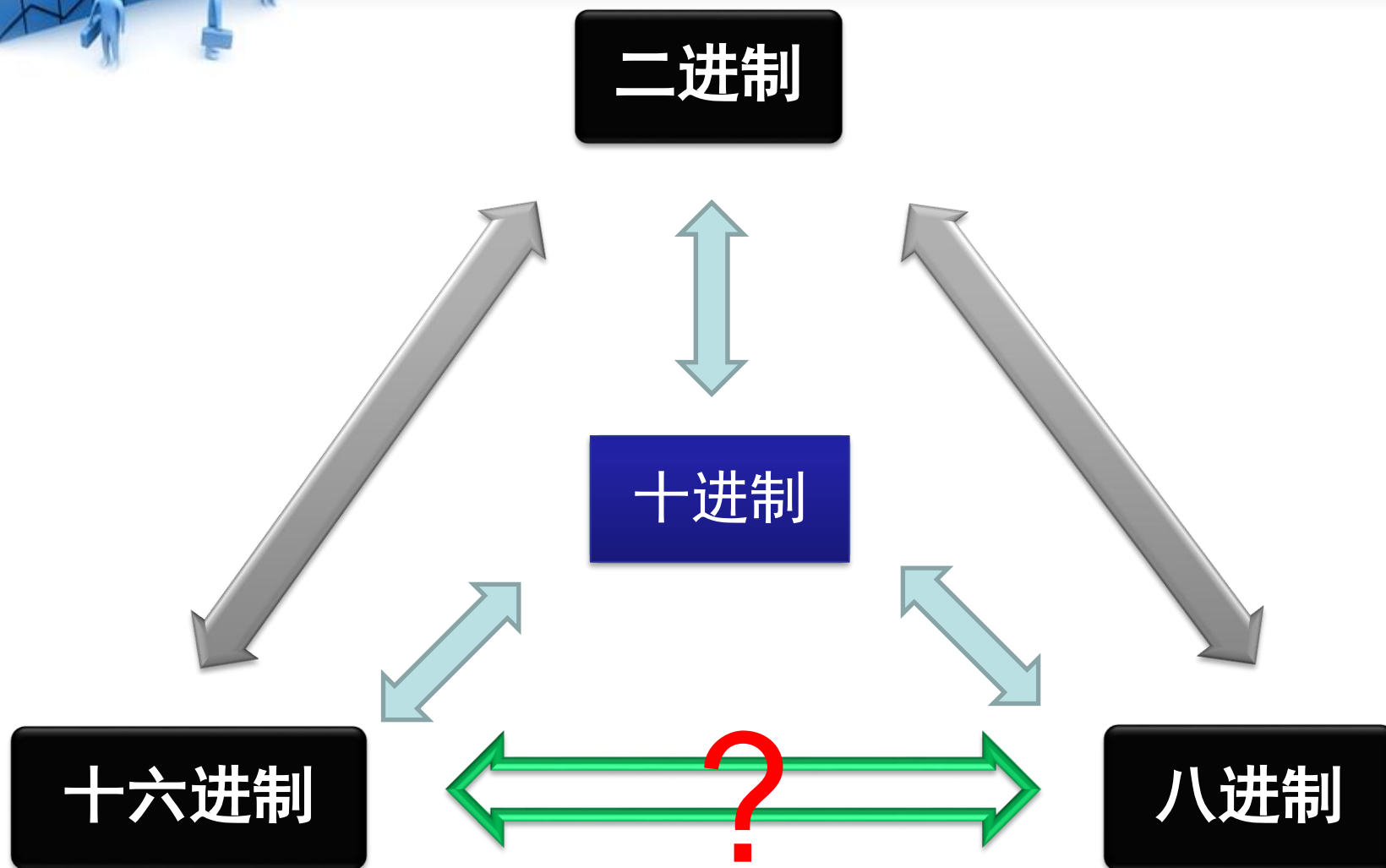
0001 0111 0011 0011

进制对照表



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	00000000	0	0	9	00001001	11	9
1	00000001	1	1	10	00001010	12	A
2	00000010	2	2	11	00001011	13	B
3	00000011	3	3	12	00001100	14	C
4	00000100	4	4	13	00001101	15	D
5	00000101	5	5	14	00001110	16	E
6	00000110	6	6	15	00001111	17	F
7	00000111	7	7	16	00010000	20	10
8	00001000	10	8	17	00010001	21	11

数制转换





- 十六进制转八进制

$$(4E2D)H = (\underline{4} \quad \underline{E} \quad \underline{2} \quad \underline{D}) = (0100111000101101)B$$
$$0100 \ 1110 \ 0010 \ 1101$$

$$(0100111000101101)B = (\underline{100} \quad \underline{111} \quad \underline{000} \quad \underline{101} \quad \underline{101})$$
$$4 \qquad 7 \qquad 0 \qquad 5 \qquad 5$$
$$= (47055)Q$$

- 八进制转十六进制

$$(4327)Q = (\underline{4} \quad \underline{3} \quad \underline{2} \quad \underline{7}) = (100011010111)B$$
$$100 \ 011 \ 010 \ 111$$

$$(100011010111)B = (\underline{1000} \quad \underline{1101} \quad \underline{0111})$$
$$8 \qquad D \qquad 7$$
$$= (8D7)H$$

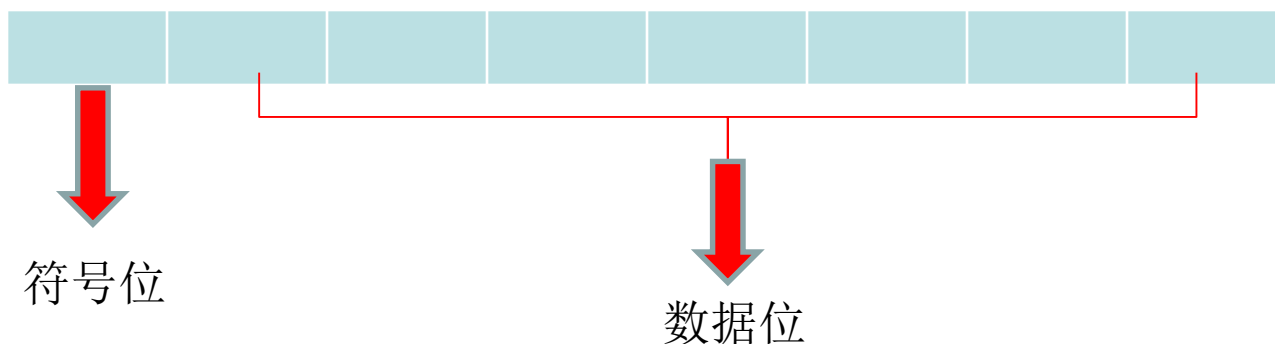


- 二进制数据的基本运算
- 二进制数据基本运算的硬件实现
- 二进制数据运算的应用
- 其他进制数据
- 不同进制数据间的转换
- 有符号数据在计算机中的存储

机器数



对于数的符号“+”或“-”，计算机是无法识别的，因此需要把数的符号数码化。通常，约定二进制数的最高位为符号位，“0”表示正号，“1”表示负号。这种在计算机中使用的表示数的形式称为**机器数**。规定：以后没有特别指明的话，规定1个字节，即8位来存储整数。





例如：

$$N_1 = 00001001$$

表示带符号数+9

$$N_2 = 11111001$$

根据不同的机器数表示不同的值，如：

原码时表示带符号数-121，

补码则表示-7，

反码则表示-6。



- 定义：符号占1位（最高位），0表示正、1表示负；数值部分按二进制书写（占剩下的位置）。
- 例如，+11
- 原码：00001011
- 例如，-11
- 原码：10001011



- 根据定义，0有两种表示方法： $+0$ ， -0 。

00000000

10000000

- 1个字节能表示 -127 到 $+127$ 这255个数

最大数：01111111

最小数：11111111



- 定义：如果是正数，与原码相同；如果是负数，符号位不变，数据位取反，末位加1。
- 例如，+11
- 补码：00001011
- 例如，-11
- 补码：11110101



- 根据定义，0只有一种表示方法： 00000000
- 特殊值-128表示方法： 10000000
- 1个字节能表示-128到+127这256个数

最大数：01111111

最小数：10000000



- 定义：如果是正数，与原码相同；如果是负数，符号位不变，数据位取反。
- 例如，+11
- 反码：00001011
- 例如，-11
- 反码：11110100



- 根据定义，0有两种表示方法： $+0$ ， -0 。

00000000

11111111

- 1个字节能表示 -127 到 $+127$ 这255个数

最大数：01111111

最小数：10000000

机器数总结

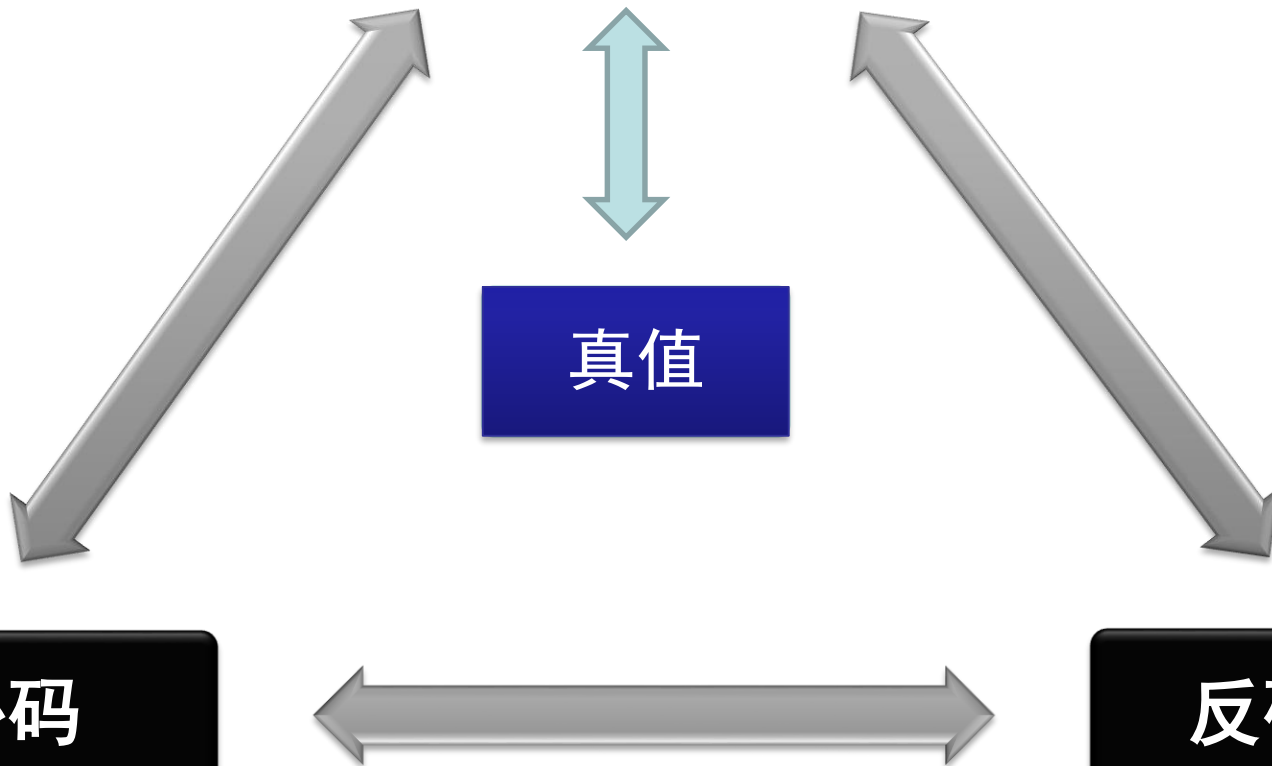


原码

真值

补码

反码



练习



- 写出下列真值的原码、反码、补码。
- -33
- -56
- 40

Questions?

