计算机中数的表示和编码(自学)

一、N进制数
- 1．N进制数的表示法
- 2．N进制数与十进制数的转换
- 3．二进制与十六进制数的转换

二、二进制数和十六进制数运算
- 1．算术运算
- 2．逻辑运算

三、计算机内数的表示
- 1．无符号数
- 2．带符号数
  - 1）求补运算
  - 2）补码
  - 3）补码的真值计算
  - 4）用补码表示带符号数的意义
- 3．8位、16位数的表示范围
- 4．进位、借位、溢出的判断

四、二进制编码
- ASCII (美国标准信息交换码)

计算机中数的表示和编码

一、N进制数

二、二进制数和十六进制数运算

三、计算机内数的表示

四、二进制编码

一、N进制数

1. N进制数的表示法
2. N进制数与十进制数的转换
3. 二进制与十六进制数的转换

1. N进制数的表示法

- 十进制数

  基数10， 遵循逢10进位

  数码10个：0，1，2，3，4，5，6，7，8，9

  如： $(123.5)_{10}$ 或123.5D 或 123.5

  数值大小计算:

  $123.5 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 5 \times 10^{-1}$

■ N进制数

基数N， 遵循逢N进位

数码N个：0，1，2，、、、，N-1

数值大小计算:

$(A_nA_{n-1} \ldots\ldots A_0 . A_{-1}A_{-2} \ldots\ldots A_{-m})_N$

$= A_n \times N^n + A_{n-1} \times N^{n-1} + \ldots\ldots + A_1 \times N^1 + A_0 \times N^0$

$+ A_{-1} \times N^{-1} + A_{-2} \times N^{-2} + \ldots\ldots + A_{-m} \times N^{-m}$

■ 二进制数

基数2，遵循逢2进位

数码2个：0，1

$(101101.1)_2$ 或 101101.1 B

$= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$+ 1 \times 2^{-1} = 45.5D$

■ 十六进制数

基数16，遵循逢16进位

数码16个：0，1，　、、、　，9，A，B，C，D，E，F

十六进制数	A	B	C	D	E	F
十进制数	10	11	12	13	14	15

$(BF3C.8)_{16}$　或　BF3C.8 H 或 0xBF3C.8

$=11 \times 16^3 + 15 \times 16^2 + 3 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1}$

=48956.5D

2. N进制数与十进制数的转换

1) N进制数 → 十进制数

方法: 与数值大小计算过程相同。

例：101101. 1 B $= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$

$= 45.5$ D

BF3C. 8 H $= 11 \times 16^3 + 15 \times 16^2 + 3 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1}$

$= 48956.5$ D

2) 十进制数 → N进制数

用除法和乘法完成

整数部分：除N取余，商零为止，结果先低后高

小数部分：乘N取整，到零为止，结果先高后低

例1　十进制数 → 二进制数

125.125D → 二进制数

整数部分：除N取余，商零为止，结果先低后高

		取余	
2	125		
2	62	1	低位
2	31	0	
2	15	1	
2	7	1	
2	3	1	
2	1	1	
	0	1	高位

商为 0

先低后高, 故：　125D = 111 1101B

小数部分：乘N取整，到零为止，结果先高后低

取整

0.125 x 2 = 0.25　　0　高位

0.25 x 2 = 0.5　　0　　↓

0.5 x 2 = 1.0　　1　低位

↖

小数为 0

先高后低，故：0.125D =0.001B

将整数部分和小数部分结合起来，

故：125.125D = 111 1101.001B

整数部分：除N取余，商零为止，结果先低后高

16 | 125　　取余

16 | 7　　13　低位

0　　7　高位

↖

商为 0

故：125D = 7DH

小数部分：乘N取整，到零为止，结果先高后低

取整

0. 125 x 16 = 2.0　　2

↖

小数为 0

故：0.125D = 0.2H

将整数部分和小数部分结合，125.125D =7D.2H

例2　十进制数 → 十六进制数

125. 125D → 十六进制数

若小数部分永不为零，可取近似值。

例　0.7 D = ? H

0.7 x 16 = 11.2

0.2 x 16 = 3.2

0.2 x 16 = 3.2

故　0.7D = 0.B333H

3. 二进制数与十六进制数的转换

1) 二进制数与十六进制数间的关系

2) 二进制数 → 十六进制数

3) 十六进制数 → 二进制数

1) 二进制数与十六进制数间的关系

十六进制数的基数 $16 = 2^4$

1位十六进制数对应4位二进制数

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

2) 二进制数 → 十六进制数

方法：

以小数点为基准，分别向左和向右每4位划为一组，不足4位用0补，每一组用其对应的十六进制数代替。

例：

11110. 01B = 0001 1110. 0100 B

= 1 E . 4 H

1111101. 001B = 0111 1101. 0010 B

= 7 D . 2 H

3) 十六进制数 → 二进制数

方法：

将每位十六进制数用其对应的4位二进制数代替即可。

例：

1E. 4H = 0001 1110. 0100B = 11110. 01B

7D. 2H = 0111 1101. 0010B = 111 1101. 001B

思考：计算机采用二进制形式表示数据和指令，
在书写，显示上引进十六进制的意义是什么？
计算机内部使用十六进制吗？

- 十进制数与二进制数之间的转换需计算，不直观；
- 二进制表示的数位多不便于书写、阅读；
- 十六进制数与二进制数间转换方便、直观，

相对于二进制数，十六进制数书写、阅读相对方便。

二、二进制数和十六进制数运算

1. 算术运算

二进制数和十六进制数加、减、乘、除, 与十进制数类似

2. 逻辑运算

二进制数运算，与、或、非、异或，特点：按位进行

1. 算术运算

1）二进制数　　加法 逢2进1
减法 借1为2

例 二进制运算 (以8位运算器为例）

```
   10110101          01000100
+  10001111       -  10100101
进位 1 111111     借位 1 111111
   01000100          10011111
```

8位运算器：参加运算的数及结果均以8位表示，
最高位产生的进位或借位在8位运算器中不保存，
而将其保存到标志寄存器中

2) 十六进制数　　加法 逢16进1
减法 借1为16

例 十六进制运算

```
   B5H          44H
+  8FH       -  A5H
进位 11     借位 11
   44H          9FH
```

结果与二进制计算相同：

```
   10110101B          01000100B
+  10001111B       -  10100101B
进位 1 111111     借位 1 111111
   01000100B          10011111B
```

三、计算机内数的表示

1. 无符号数
2. 带符号数
   1) 求补运算
   2) 补码
   3) 补码的真值计算
   4) 用补码表示带符号数的意义
3. 8位、16位数的表示范围
4. 进位、借位、溢出的判断

1. 无符号数

二进制数的各位均表示数值大小，最高位无符号意义。

例　1111 0000 B = F0H = 15x16 = 240D

1001 0001 B = 91H = 9x16 + 1= 145 D

应用场合：

处理的数全是正数时，如表示地址的数

2. 带符号数

- 数有正、负 → 带符号数
- 在计算机中符号也用二进制数表示
- 计算机中用补码表示带符号数

1) 求补运算

对一个二进制数按位取反，最低位加1。

等价于：0 - 该二进制数

即用0 减去该二进制数

例：对 8 位二进制数 11110001B进行求补运算

方法1：按位取反，最低位加1

```
        1111 0001 B
取反    0000 1110 B
加1          1
-------------------
        0000 1111 B
```

方法2： 0 - 该二进制数

```
                0000 0000 B        0 0 H
-               1111 0001 B      - F 1 H
-------------------------------------------
                0000 1111 B        0 F H
```

最高位借位超出8位自然丢失

2) 补码

在计算机中，用补码表示带符号数。

补码的表示方法：

正数的补码：最高位为 0，

其它各位为数字位，表示数的大小。

负数的补码：通过对该数正数的补码进行求补运算得到。

负数的补码最高位为 1。

例　求 105D 的补码

```
2 |105
  2 |52      1
  2 |26      0
  2 |13      0
  2 | 6      1
  2 | 3      0
  2 | 1      1
    0        1
```

$[105D]_{补}$

= 0110 1001B = 69 H　　(8位)

= 0000 0000 0110 1001 B= 0069 H　　(16位)

正数的补码：最高位为0

其它各位为数字位，表示数的大小。

例　求 – 105D的补码

8位：$[-105D]_{补} = 0 - [105D]_{补}$

$= 0 - 0110\ 1001B = 0 - 69H$

$= 10010111B = 97\ H$

16位：$[-105D]_{补} = 0 - [105D]_{补}$

$= 0 - 0000\ 0000\ 0110\ 1001B = 0 - 0069H$

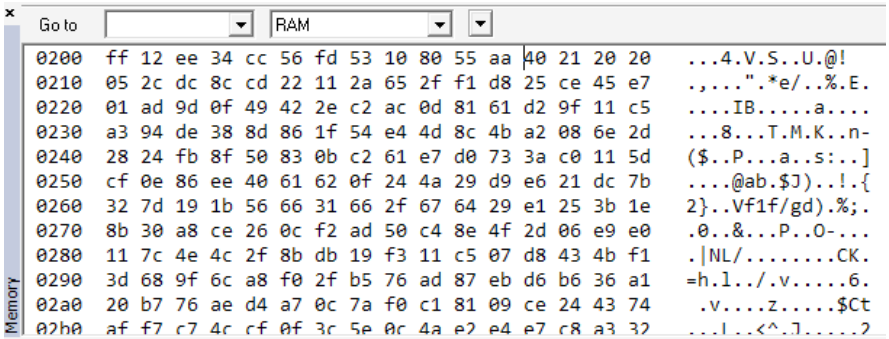
$= 1111\ 1111\ 1001\ 0111B = F\ F\ 9\ 7\ H$

负数的补码：通过对该数正数的补码进行求补运算得到。

3) 补码的真值计算

真值：补码表示的数值大小。

如用EW430查看到存放在内存中的一组符号数：



如何知道它们表示的大小？

求补码真值的方法：

- 先判断是正数，还是负数。

  由最高位判断：0 → 正数

  1 → 负数

- 再求数值大小

  对正数，补码的真值等于该二进制数值。

  对负数，先对该数进行求补运算，再求数值大小。

例　求补码7D H 的真值:

7D H = 0111 1101B ，　最高位为0，是正数

7DH的真值 = 7 x 16 + 13 = 125 D

例　求补码 91H 的真值:

91H = 1001 0001B，　最高位为1 ，是负数。

对91H进行求补运算：

91H $\xrightarrow{\text{求补}}$ 00H – 91H = 6F H

91H的真值 = – 6FH = – (6 x 16 +15) = – 111D

4）用补码表示带符号数的意义　　计算机中用补码表示带符号数。

(1) 将减法用加法实现，省去减法器，简化硬件。

计算机中，减法实现过程：（补码减法）

- 先对减数进行求补运算（求反加1，也是加法）
- 再将求补后的数与被减数相加
- 相加的结果即为用补码表示的两数相减结果。

计算：

$$\begin{array}{r} 01000100B \\ -\ 10100101B \\ \hline \end{array}$$

计算机中实现过程：

$$\begin{array}{r} 01011010 \\ +\qquad 1 \\ \hline 01011011 \\ +\ 01000100 \\ \hline 10011111 \end{array}$$

计算：

$$\begin{array}{r} 01000100B \\ -\ 10100101B \\ \hline \end{array}$$

补码减法：

$$\begin{array}{r} 01011010 \\ +\qquad 1 \\ \hline 01011011 \\ +\ 01000100 \\ \hline 10011111 \end{array}$$

常规减法：

$$\begin{array}{rl} & 01000100\ B \\ -\ & 10100101\ B \\ \text{借位}\ 1 & \ 111111 \\ \hline & 10011111\ B \end{array}$$

▲补码减法的计算结果与常规减法的结果相同。

▲人在计算时，可用常规减法
(补码减法，对人来说，相对复杂)

2) 无符号数及带符号数的加减运算用同一电路完成。

即指令系统中加、减运算不区分无符号数或带符号数。

例：8位运算器

在计算机中计算	看作无符号数	看作带符号数
1111 0001 + 0000 1100 1111 1101	241 + 12 253	(−15) + (+12) − 3
1111 0001 − 0000 1100 1110 0101	241 − 12 229	(−15) − (+12) −27

3. 8位、16位二进制数的表示范围

	无符号数	带符号数
8 位	0 ~ 255	−128 ~ 127
16 位	0 ~ 65535	−32768 ~ 32767

规定:

8位 1000 0000B 即 80H为 – 128D

16位 1000 0000 0000 0000B 即 8000H 为 – 32768D

4. 进位、借位、溢出的判断

1) 进位

在加法过程中，最高有效位向高位产生进位。

对 8位运算，指D7产生进位

对16位运算，指D15产生进位

2) 借位

在减法过程中，最高有效位向高位产生借位。

对 8位运算，指D7产生借位

对16位运算，指D15产生借位

3) 溢出

指加减运算结果超出带符号数表示的范围。

8位 -128 ~127

16位 -32768 ~ 32767

溢出的判断方法：

由参与运算的两数及其结果的符号位进行判断，结论：

- 符号相同的两数相加，
  所得结果的符号与之相反，结果溢出。
- 符号相异的两数相减，
  所得结果的符号与减数相同，结果溢出。
- 其他情况，不会产生溢出。
  符号相异的两数相加，或符号相同的两数相减

产生溢出的情况
正数+正数=负数
负数+负数=正数
正数-负数=负数
负数-正数=正数

注意：进位、借位与溢出的区别。

例：

加法

```
  0111 1111
+ 0000 0010
-----------
  1000 0001
```

→

补码真值

```
   127
+    2
------
  -127
```

正 + 正 ⟶负，无进位，产生溢出，表示结果错

```
  1000 0001
+ 1000 0010
-----------
  0000 0011
```

→

```
  -127
+ -126
------
   + 3
```

进位保存在标志中

负 + 负 ⟶正，有进位，产生溢出，表示结果错

例：　　减法　　补码真值

```
    0000 0010          2
-   11000 0001    →    -    -127
    1000 0001          -127
```

借位保存在标志中

正 − 负 ⟶负，有借位，产生溢出，表示结果错

```
    1000 0001          -127
-   0000 0010    →     -      2
    0111 1111          +127
```

负 − 正 ⟶正，无借位，产生溢出，表示结果错

四、二进制编码

- 计算机处理的信息：数值、字符(字母、汉字等)
- 各字符在计算机中由若干位的二进制数表示
- 二进制数与字符之间一一对应的关系，称字符的二进制编码。

ASCII（美国标准信息交换码）

微机中普遍采用的字符编码，如键盘、打印机、显示器等

	ASCII
数字 0～9	30H～39H
小写 a～z	61H～7AH
大写 A～Z	41H～5AH
回车符	0DH
换行符	0AH
空格	20H

例　在EW430下查看字符串在存储器的内容：

用16进制表示存储器单元的地址和内容，内容看作ASCII对应的字符