

## 第2节 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)

## 第二节 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：

$$(A_n A_{n-1} \dots A_0 . A_{-1} A_{-2} \dots A_{-m})_N$$

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

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

## ● 二进制数

基数**2**，遵循逢**2**进位

数码**2**个：**0**，**1**

$(101101.1)_2$  或  $101101.1\text{ B}$

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

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

## ● 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)<sub>16</sub> 或 BF3C. 8 H

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

$$=48956. 5D$$

### 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  $\rightarrow$  十六进制数

3) 十六进制数  $\rightarrow$  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:

$$\begin{aligned} 11110.01\text{B} &= \underline{0001} \ \underline{1110}. \ \underline{0100} \ \text{B} \\ &= 1 \quad \text{E} . \ 4 \ \text{H} \end{aligned}$$

$$\begin{aligned} 1111101.001\text{B} &= \underline{0111} \ \underline{1101}. \ \underline{0010} \ \text{B} \\ &= 7 \quad \text{D} . \ 2 \ \text{H} \end{aligned}$$

### 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。

# 在EW430下查看存储器单元和CPU寄存器的内容

Goto		RAM	
0300	33 a0 ef 4e 63 89 39 02 26 00 0f ee 41 73 22 be		3..Nc.9.&...As".
0310	cc a4 d3 95 b6 e7 89 73 c1 02 10 c8 c2 37 93 75		.....s.....7.u
0320	c0 be c6 89 43 28 4f 6a e2 f2 83 50 7b 5b 94 2a		....C(0j...P{[.*
0330	f7 a8 be 5b b4 f4 76 f4 e3 ba d7 48 e4 26 b7 7b		...[..v....H.&.{
0340	13 f7 98 04 66 f4 67 ab fb 23 cd 82 4b e5 71 de		....f.g..#..K.q.
0350	35 33 6b 53 ab 63 d7 44 91 6f 45 f2 1c 85 6a 2d		53kS.c.D.oE...j-
0360	d9 5a 56 81 a0 9c 19 1e 09 65 91 b8 b9 96 50 32		.ZV.....e....P2
0370	a3 fc 50 bd 7e 2c e8 d3 16 60 42 36 46 e3 a5 bb		..P.~,...`B6F...
0380	aa c7 78 bc 65 e0 41 ca 89 e0 78 19 76 03 0f 2c		..x.e.A...x.v...,
0390	4c 14 e6 cb 33 d9 a8 c6 21 9d b7 70 e1 60 a8 08		L...3...!..p.`..
03a0	7d 77 7a 61 cd 16 80 73 5c 11 af 3b d2 55 cf 09		}wza...s\...;.U..
03b0	18 57 2d a7 f5 0a d6 78 46 8d 14 76 14 af f6 2a		.W-....xF...v...*

用**16**进制表示存储器单元的地址和内容

用**16**进制表示寄存器的内容

Register	
CPU Registers	
PC	= 0x1114
SP	= 0x0A00
SR	= 0x0005
R4	= 0xFFFF
R5	= 0x7D9C
R6	= 0x3604
R7	= 0x2112
R8	= 0x4AC6
R9	= 0x5601
R10	= 0x007A
R11	= 0x2D04
R12	= 0x323A
R13	= 0x1609
R14	= 0x1473
R15	= 0x4F24

## 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 (第二章逻辑指令中介绍)

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)

$$\begin{array}{r} 10110101 \\ + 10001111 \\ \hline \text{进位 } 1 \quad 111111 \\ \hline 01000100 \end{array}$$

$$\begin{array}{r} 01000100 \\ - 10100101 \\ \hline \text{借位 } 1 \quad 111111 \\ \hline 10011111 \end{array}$$

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

$$\begin{array}{r}
 \text{B 5 H} \\
 + \text{8 F H} \\
 \hline
 \text{进位 1 1} \\
 \hline
 \text{4 4 H}
 \end{array}$$

$$\begin{array}{r}
 \text{4 4 H} \\
 - \text{A 5 H} \\
 \hline
 \text{借位 1 1} \\
 \hline
 \text{9 F H}
 \end{array}$$

结果与二进制计算相同:

$$\begin{array}{r}
 \text{1 0 1 1 0 1 0 1 B} \\
 + \text{1 0 0 0 1 1 1 1 B} \\
 \hline
 \text{进位 1 1 1 1 1 1} \\
 \hline
 \text{0 1 0 0 0 1 0 0 B}
 \end{array}$$

$$\begin{array}{r}
 \text{0 1 0 0 0 1 0 0 B} \\
 - \text{1 0 1 0 0 1 0 1 B} \\
 \hline
 \text{借位 1 1 1 1 1 1} \\
 \hline
 \text{1 0 0 1 1 1 1 1 B}
 \end{array}$$



## 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  $1111\ 0000\ \text{B} = \text{F0H} = 15 \times 16 = 240\text{D}$

$$1001\ 0001\ \text{B} = 91\text{H} = 9 \times 16 + 1 = 145\ \text{D}$$

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： $0 - \text{该二进制数}$



## 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

$$2 \overline{)105}$$

$$2 \overline{)52}$$

$$2 \overline{)26}$$

$$2 \overline{)13}$$

$$2 \overline{)6}$$

$$2 \overline{)3}$$

$$2 \overline{)1}$$

0

1

0

0

1

0

1

1



[ 105D ]<sub>补</sub>

$$= \underline{0110\ 1001} \text{B} = 69 \text{ H} \quad (8\text{位})$$

$$= \underline{0000\ 0000\ 0110\ 1001} \text{B} = 0069 \text{ H} \quad (16\text{位})$$

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

$$\begin{aligned} \text{8位: } [-105\text{D}]_{\text{补}} &= 0 - [105\text{D}]_{\text{补}} \\ &= 0 - 0110\ 1001\text{B} && \Big| = 0 - 69\text{H} \\ &= 10010111\text{B} && \Big| = 97\ \text{H} \end{aligned}$$

$$\begin{aligned} \text{16位: } [-105\text{D}]_{\text{补}} &= 0 - [105\text{D}]_{\text{补}} \\ &= 0 - 0000\ 0000\ 0110\ 1001\text{B} && \Big| = 0 - 0069\text{H} \\ &= 1111\ 1111\ 1001\ 0111\text{B} && \Big| = \text{FF}97\ \text{H} \end{aligned}$$

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：

Go to	RAM	
0200	ff 12 ee 34 cc 56 fd 53 10 80 55 aa	40 21 20 20 ...4.V.S..U.@!
0210	05 2c dc 8c cd 22 11 2a 65 2f f1 d8 25 ce 45 e7	.,...".*e/..%.E.
0220	01 ad 9d 0f 49 42 2e c2 ac 0d 81 61 d2 9f 11 c5	....IB.....a....
0230	a3 94 de 38 8d 86 1f 54 e4 4d 8c 4b a2 08 6e 2d	...8...T.M.K..n-
0240	28 24 fb 8f 50 83 0b c2 61 e7 d0 73 3a c0 11 5d	(\$..P...a..s:..]
0250	cf 0e 86 ee 40 61 62 0f 24 4a 29 d9 e6 21 dc 7b	....@ab.\$J)..!.{
0260	32 7d 19 1b 56 66 31 66 2f 67 64 29 e1 25 3b 1e	2}..Vf1f/gd).%;. .
0270	8b 30 a8 ce 26 0c f2 ad 50 c4 8e 4f 2d 06 e9 e0	.0..&...P..0-...
0280	11 7c 4e 4c 2f 8b db 19 f3 11 c5 07 d8 43 4b f1	. NL/.....CK.
0290	3d 68 9f 6c a8 f0 2f b5 76 ad 87 eb d6 b6 36 a1	=h.1../.v.....6.
02a0	20 b7 76 ae d4 a7 0c 7a f0 c1 81 09 ce 24 43 74	.v....z.....\$Ct
02b0	af f7 c7 4c cf 0f 3c 5e 0c 4a e2 e4 e7 c8 a3 32	...l...<^..J.....?

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： $0 \rightarrow$  正数

$1 \rightarrow$  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 =  $7 \times 16 + 13 = 125 \text{ D}$

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 =  $-6FH = -(6 \times 16 + 15) = -111D$

## 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：

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

计算机中实现过程：

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

计算:

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

补码减法:

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

常规减法:

$$\begin{array}{r} 01000100\text{ B} \\ - 10100101\text{ B} \\ \hline \text{借位 } 1 \quad 11111 \\ \hline 10011111\text{ B} \end{array}$$

▲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
$\begin{array}{r} 1111\ 0001 \\ +\ 0000\ 1100 \\ \hline 1111\ 1101 \end{array}$	$\begin{array}{r} 241 \\ +\ 12 \\ \hline 253 \end{array}$	$\begin{array}{r} (-15) \\ +\ (+12) \\ \hline -3 \end{array}$
$\begin{array}{r} 1111\ 0001 \\ -\ 0000\ 1100 \\ \hline 1110\ 0101 \end{array}$	$\begin{array}{r} 241 \\ -\ 12 \\ \hline 229 \end{array}$	$\begin{array}{r} (-15) \\ -\ (+12) \\ \hline -27 \end{array}$

### 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。

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

符号相异的两数相加，或符号相同的两数相减

注意：进位、借位与溢出的区别。

例：加法

补码真值

$$\begin{array}{r} 0111\ 1111 \\ + 0000\ 0010 \\ \hline 1000\ 0001 \end{array}$$



$$\begin{array}{r} 127 \\ + 2 \\ \hline -127 \end{array}$$

正 + 正  $\rightarrow$  负，无进位，产生溢出，结果错

$$\begin{array}{r} 1000\ 0001 \\ + 11000\ 0010 \\ \hline 0000\ 0011 \end{array}$$



$$\begin{array}{r} -127 \\ + -126 \\ \hline + 3 \end{array}$$

进位保存在标志中

负 + 负  $\rightarrow$  正，有进位，产生溢出，结果错

# 例： 减法

# 补码真值

$$\begin{array}{r} 0000\ 0010 \\ - 11000\ 0001 \\ \hline 1000\ 0001 \end{array}$$



$$\begin{array}{r} 2 \\ - -127 \\ \hline -127 \end{array}$$

借位保存在标志中

正 - 负 → 负，有借位，产生溢出，结果错

$$\begin{array}{r} 1000\ 0001 \\ - 0000\ 0010 \\ \hline 0111\ 1111 \end{array}$$



$$\begin{array}{r} -127 \\ - 2 \\ \hline +127 \end{array}$$

负 - 正 → 正，无借位，产生溢出，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:

x		Goto		RAM														
	0200	48	65	6c	6c	6f	20	4d	53	50	34	33	30	40	21	20	20	Hello MSP430@!
	0210	05	2c	dc	8c	cd	22	11	2a	65	2f	f1	d8	25	ce	45	e7	.,...".*e/..%.E.
	0220	01	ad	9d	0f	49	42	2e	c2	ac	0d	81	61	d2	9f	11	c5	....IB.....a....
	0230	a3	94	de	38	8d	86	1f	54	e4	4d	8c	4b	a2	08	6e	2d	...8...T.M.K..n-
	0240	28	24	fb	8f	50	83	0b	c2	61	e7	d0	73	3a	c0	11	5d	(\$..P...a..s:...]
	0250	cf	0e	86	ee	40	61	62	0f	24	4a	29	d9	e6	21	dc	7b	....@ab.\$J)..!.{
	0260	32	7d	19	1b	56	66	31	66	2f	67	64	29	e1	25	3b	1e	2}..Vf1f/gd).%;
	0270	8b	30	a8	ce	26	0c	f2	ad	50	c4	8e	4f	2d	06	e9	e0	.0..&...P..O-...
	0280	11	7c	4e	4c	2f	8b	db	19	f3	11	c5	07	d8	43	4b	f1	. NL/.....CK.
Memory	0290	3d	68	9f	6c	a8	f0	2f	b5	76	ad	87	eb	d6	b6	36	a1	=h.l../.v.....6.
	02a0	20	b7	76	ae	d4	a7	0c	7a	f0	c1	81	09	ce	24	43	74	.v....z.....\$Ct
	02b0	af	f7	c7	4c	cf	0f	3c	5e	0c	4a	e2	e4	e7	c8	a3	32	...l...<^..].....2