习题

1. 计算机系统中A/D、D/A器件的作用是什么？

A/D实现模拟量转换为数值量，D/A实现数值量到模拟量的转换。

1. 8位十进制整数，用多少位二进制整数表示？

解：因为 108=2*x*

ln108=ln2*x*

*x*=8ln10/ln2

≈26.575

取大于*x*的最小整数27，所以用27位二进制整数表示。

3．将下面各十进制数分别转化为二进制、八进制和十六进制数。

①1023=210-1=（1111111111）2=（1 111 111 111）2=（1777）8=（11 1111 1111）2=3FFH，

②53=32+16+5=（110 101）2=（65）8=（0011 0101）2=35H

③119.25

119=96+23=64+32+16+7=（001 110 111）2=（167）8=（0111 0111）2=77H

0.25=0.010=(0.2)8=(0.0100)16

119.25=（167.28=(77.4)16

④63=48+15=32+16+15=(111 111)2=（77)8=3FH

⑤512=29=(1000000000)2=(1000)8=(0010 0000 0000)2=(2 0 0)16

⑥255=28-1=（11 111 111）2=(377)8=（1111 1111）2=(FF)16

⑦555=29+32+11=（1 000 101 011）2=(1053)8=（10 0010 1011）2=(22B)16

⑧324=28+64+4=（101 000 100）2=(504)8=（1 0100 0100）2==(144)16

4．将下面十六进制数转化为二进制、八进制和十进制数。

①FFEH=（111 111 111 110）2=(7776)8=163-2=4096-2=4094

②123H=（000 100 100 011）2=(443)8=1×162+2×161+3=256+19=291

③56H=（01 010 110）2=(126)8=5×161+6=86

④5E8H=（010 111 101 000）2=(2750)8=5×162+14×161+8=1512

67.01H=（01100111.000 000 010）2=(147.002)8=96+7+16-2=103.00390625

1. 将10110110.0111B分别转化为八进制和十六进制数。

10 110 110.011 1B==(266.34)8=1011 0110.011 1B=(B6.7)16

6．数制转换

(78．8)16 =(？)10 ， (121.02)3 =(？)4 。

解：(78．8)16 =7\*16+8+8\*16-1=112+8+0.5=120.5

(121.02)3 =1\*32+2\*3+1+2\*3-2=9+6+1+2/9=16.222222……=(100.032032……)4

1. 已知[*x*] 补=1,01101，求[*x*]原和[*x*]反 。

解：已知[x]补=1,01101

[x]原=1,10011

[x]反=1,01100

8．采用两个字节写出下列各数的原码、反码和补码。

*x* = +0.1011， [*x*]补=0.101100000000000， [*x*]原=0.101100000000000，

[*x*]反=0.101100000000000

*x* =-0.1101， [*x*]补=1.001100000000000， [*x*]原=1.110100000000000

[*x*]反=1.001011111111111

*x* = +0.1111 [*x*]补=0.111100000000000， [*x*]原=0.111100000000000，

[*x*]反=0.111100000000000

*x* =-0.0101， [*x*]补=1.101100000000000， [*x*]原=1.010100000000000，

[*x*]反=1.101011111111111

*x* =+111001， [*x*]补=0,000000000111001， [*x*]原=0,000000000111001，

[*x*]反=0,000000000111001

*x* =-110011， [*x*]补=1,111111111001101， [*x*]原=1,000000000110011，

[*x*]反=1,111111111001100

*x* =+100111， [*x*]补=0,000000000100111， [*x*]原=0,000000000100111，

[*x*]反=0,000000000100111

*x* =-000111， [*x*]补=1,111111111111001， [*x*]原=1,000000000000111，

[*x*]反=1,111111111111000

9．已知下列二进制数，用二进制运算规则计算：A+B，A-B，C×D，C÷D

解：已知A=10110100，B=1011110，C=1010101，D=110

A+B=100010010，

10110100

+ 1011110

100010010

1. B=1010110，

10110100

- 101111 0

1010110

C×D=10000100011000

1010101

× 11 0

1010101

1010101

111111 110

C÷D=1110+1/110

111 0

110 1010101

110

1001

110

110

110

1

1. 位、字节、字、字长的含义是什么？

位（bit）：二进制数的一个数位的数字；字节：八个bit的二进制数据；字：计算机运算器可直接计算和数据线可直接传输的位数，体现计算机的性能；字长：计算机字的长度。

1. ASCII码的作用是什么？对英文字符编码。

12．采用8421BCD码计算345+789=？

解：345=( 0011 0100 0101)8421BCD，789=(0111 1000 1001)8421BCD

0011 0100 0101

+ 01111 10001 1001

1011 1101 1110

+ 1 0110 0110 0110

0001 0001 0011 0100

( 0011 0100 0101)8421BCD+（0111 1000 1001)8421BCD=(0001 0001 0011 0100)8421BCD

13．设[*x*]补=0,11001，[*y*]补=1,00111，求[*x*/2]补、 [*y*/2]补、 [*x*/8]补、 [*y*/8]补、[-*x*]补 、[-*y*]补。

解：已知[x]补=0,11001

[x/2]补=0,01100 符号不变，右移一位，保持原位数n+1不变

[x/4]补=0,00110 符号不变，右移一位，保持原位数n+1不变

[x/8]补=0,00011

[y]补=1,00111

[y/2]补= 1,10011

[y/4]补= 1,11001 符号不变，右移一位，保持原位数n+1不变

[y/8]补= 1,11100 符号不变，右移一位，保持原位数n+1不变

已知[x]补= 0,11001

[-x]补=1,00111 连同符号位取反加1

已知[y]补= 1,00111

[-y]补=0,11001 连同符号位取反加1

1. 已知x=+1101，y=-1011，试用变形补码表示法完成x+y和x-y运算，并判断溢出。

解：已知x=+1101，y=-1011，

有 [x]补= 00,1101 ，[y]补=11,0101 ，[-y]补=00,1011，[-x]补=11,0011

00,1101 00,1101 11,0101

+ 11,0101 + 00,1011 + 11,0011

00,0010 01,1000 10,1000

[x+y]补=00,0010 [x-y]补=01,1000上溢出 [y-x]补=[y]补+[-x]补 =10,1000下溢出

1. 设二进制浮点数的阶码四位，尾数八位（含符号位），当采用原码表示时，写出它的规格化最大正数、最小正数、最大负数、最小负数，并写出相应的十进制的数值。

解：已知E=EsE1E2E3，M=Ns。N1N2N3N4N5N6N7

规格化最大正数M=Ns。N1N2N3N4N5N6N7=0.1111111，E=EsE1E2E3=0,111

规格化最小正数M=Ns。N1N2N3N4N5N6N7=0.1000000，E=EsE1E2E3=1,111

规格化最大负数M=Ns。N1N2N3N4N5N6N7=1.1000000，E=EsE1E2E3=1,111

规格化最小正数M=Ns。N1N2N3N4N5N6N7=1.1111111，E=EsE1E2E3=0,111

16.查表得字符'0'和'a'的ASCII码，可以据此推出字符'7'和和'f'的ASCII码吗？

解：'0'=30H，'7'=30H+7=37H

'a'=61H，'f'=61H+5=66H

17．（选做）32位IEEE754标准的浮点数格式为：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ns | Es | E1 | E2 | …… | En | N-1 | N-2 | …… | N-m |

数符 阶码 小数点 尾数

其中数的符号位1位；阶码8位，用移码表示，偏移量为127；尾数M占低23位，用原码表示，小数点在尾数域的最前面，尾数规格化时，将小数点左边保持为1，并默认省略，使得23位尾数起到24位效果。将十进制数28.75转换为32位IEEE754短浮点数。

解：28.75=（11100.11）2 =（1.110011）2 ×200000100

M=0.11001100000000000000000

E=00000100+01111111=10000011

(28.75)IEEE754短浮点数=010000011 11001100000000000000000

附加题1

一、转换下列各数，转换中分别用截断法或四舍五入法保留五位二进制小数计算。

(1112)3=(41)10=( 131 )5

(FC)16=( 374)8=(1111 1100)2

(-0.232)10= ( -0.0011101 )2= ( -0.0322 )4

(11012.321)4=(146.E4 )16

(87023.721)9=( 2221000210.210201 )3

二、计算题

1已知 *x*=+0.10101，*y*=-0.01101，求[*x*]补+[*y*]补  （机器数为双字节）

解：

*x*=+0.10101,[*x*]补= 00.10101000000000

*y*=-0.01101，[*y*]补= 11.10011000000000

[*x*]补+[*y*]补 = 00. 01000000000000

2.已知 *x*=+101011，*y*=+101，求*x*×*y*和*x*/*y*

解：*x*×*y=*101011×101=11010111 *x*/*y* =1000+11/101

1000

101011 101 101011

× 101 101

101011 011

101011

11010111

3.已知[*x*]补=1.1101011，求[*x*/2]补，[*x*/8]补，[-*x*]补。（字长为8位，包含符号位）

解：

[*x*]补 =1.1101011

[*x*/2]补 =1.1110101

[*x*/8]补 =1.1111101

[*x*]补 =1.1101011

[-*x*]补 =0.0010101

4.已知*x*=-0.10011，*y*=0.00101，求[*x*+*y*]反

解： *x*=-0.10011， [*x*]反=1.01100

*y*= 0.00101， [*y*]反=0.00101

[*x*+*y*]反=1.10001

有 *x*+*y*= -0.01110

5.已知[*x*]补=1,1110101，求*x*、[*x*]原

解： [*x*]补=1,1110101，则*x=*-0001011，[*x*]原 =1,0001011

1. 已知[*x*] 原=1,1110101，求*x*、[*x*]补

解： [*x*] 原=1,1110101，则*x=*-1110101，[*x*]补 =1,0001011

7.已知[*x*] 反=1,1110111，求*x*、[*x*]补

解： [*x*] 反=1,1110111，则*x=*-0001000，[*x*]补 =1,1111000

8.已知*x*=-0.56、y=-0.625、z=0.75采用n+2=9补码计算二进制的*x+y*和*x+z*代数和，判断计算过程中是否溢出？

解：已知*x*=-0.56=(-0.1001000)2，[*x*]补 =11.0111000

-0.56 -0.12 -0.24 -0.48 -0.96 -0.92 -0.84 -0.68 -0.36 -0.72……

1 0 0 0 1 1 1 1 0 1

y=-0.625=（-0.1010000)2，[*y*]补 =11.0110000

*z*=0.75=（0.1100000)2，[*z*]补 =00.1100000

[*x+y*]补=11.0111000+11.0110000=

11.011 1000

+ 11.011 0000

10 .110 1000 负溢出

[*x+z*]补=11.0111000+00.1100000=00.001 1000

11.011 1000

+ 00.110 0000

00.001 1000

附加题2

1. 采用补码表示*M*（带符号5位）、*E*（带符号5位）的浮点数如下，求浮点数真值。

求其真值。



•

阶码 数符 尾数

解：补码表示的尾数和阶码分别为：

*M=*0.0111，*E*=1,0011

它们的真值为：

m=0.0111，e=-1101

浮点数对应的真值为：

（0.0111)2×2-13

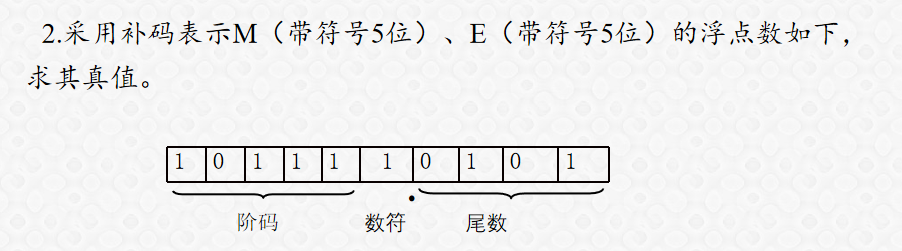
=(0.25+0.125+0.0625)×2-13

=0.4375×2-13

或

（0.0111)2×2-13

=0.000000000000001112



解：补码表示的尾数和阶码分别为：

*M=*1.0101，*E*=1,0111

它们的真值为：

m=-0.1011，e=-1001

浮点数对应的真值为：

（-0.1011)2×2-9

=-(0.5+0.125+0.0625)×2-9

=-0.6875×2-9

或

（-0.1011)2×2-9

=-0.00000000010112

1. 浮点数尾数*M*（带符号16位）和阶码*E*（带符号8位）采用补码，试表示实数56.625和-778.66。

解：

56.625=（ 111000.101)2=(0. 111000101)2×2+6

*M*=0. 1110001010000000，*E*=0,0000110

-778.66=（-1100001010.101010)2=（-0.110000101010101)2×2+10 尽量保留有效数字

*M*=1.001111010101011， *E*=0,0001010

.66 0.32 0.64 0.28 0.56 0.12 0.24 0.48 0.96 0.92 0.84 0.68 0.36

1 0 1 0 1 0 0 0 1 1 1 1 0

1. 用补码表示阶码E（一个字节）和尾数M（24比特），求浮点数的取值范围。

解：用补码表示阶码E（一个字节）和尾数M（24比特），浮点数的取值范围

1）最大正数的求取

因为用补码表示的阶码E占用一个字节，则

E=Es E1 E2 E3 E4 E5 E6 E7

最大正数浮点数的阶码应该是可表示的最大正整数，即 E=0,1111111

因为用补码表示的尾数M占用24比特，则

M=Ms .M-1M-2M-3M-4M-5M-6M-7M-8M-9M-10M-11M-12M-13M-14M-15M-16M-17M-18M-19M-20M-21M-22M-23

最大正数浮点数的尾数应该是可表示的最大正的纯小数，即

M=0.11111111111111111111111

对应的浮点数的真值

0.111111111111111111111112×2127

=(1-2-23)×2127

2）最小正数的求取

因为用补码表示的阶码E占用一个字节，则

E=Es E1 E2 E3 E4 E5 E6 E7

最小正数浮点数的阶码应该是可表示的最小负整数，即

E=[-10000000]补=1,0000000

因为用补码表示的尾数M占用24比特，则

M=Ms .M-1M-2M-3M-4M-5M-6M-7M-8M-9M-10M-11M-12M-13M-14M-15M-16M-17M-18M-19M-20M-21M-22M-23

最小正数浮点数的尾数应该是可表示的最小正的纯小数，即

M=0.00000000000000000000001

对应的浮点数的真值

0.000000000000000000000012×2-128

=2-23×2-128

=2-151

3）最大负数的求取

最大负数与最小正数绝对值相同，符号相反。

可得补码 E=[-10000000]补=1,0000000

M=[-0.000000000000001]补=1.11111111111111111111111

对应的浮点数的真值

-0.000000000000000000000012×2-128

=-2-23×2-128

=-2-151

4）最小负数的求取

最小负数浮点数的阶码应该是可表示的最大正整数，即补码

E=[+1111111]补=0,1111111

最小负数浮点数的尾数应该是负的，绝对值最大，即补码

M=[-1.000000000000000]补=1.00000000000000000000000

浮点数对应的真值

-1.000000000000000000000002×2127

= -1×2127

1. 用原码表示阶码*E*（6比特）和尾数*M*（一个字节），求浮点数的取值范围。

1）最大正数的求取

因为用原码表示的阶码E占用6比特，则

E=Es E1 E2 E3 E4 E5

最大正数浮点数的阶码应该是可表示的最大正整数，即

E=0,11111

因为用原码表示的尾数M占用一个字节，则

M=Ms .M-1M-2M-3M-4M-5M-6M-7

最大正数浮点数的尾数应该是可表示的最大正的纯小数，即

M=0.1111111

对应的浮点数的真值

（1-2-7）×231

2）最小正数的求取

因为用原码表示的阶码E占用一个字节，

E=Es E1 E2 E3 E4 E5

最小正数浮点数的阶码应该是可表示的最小负整数，

E=1,1111111

因为用原码表示的尾数M占用两个字节，则

M=Ms .M-1M-2M-3M-4M-5M-6M-7

最小正数浮点数的尾数应该是可表示的最小正的纯小数，即

M=0.0000001

浮点数对应的真值

0.0000001×2-31

=2-7×2-31

=2-38

3）最大负数的求取

最大负数与最小正数绝对值相同，符号相反。

可得原码E=1,11111和 M=1.0000001，对应的浮点数的真值

-0.0000001×2-31

=2-7×2-31

=2-38

4）最小负数的求取

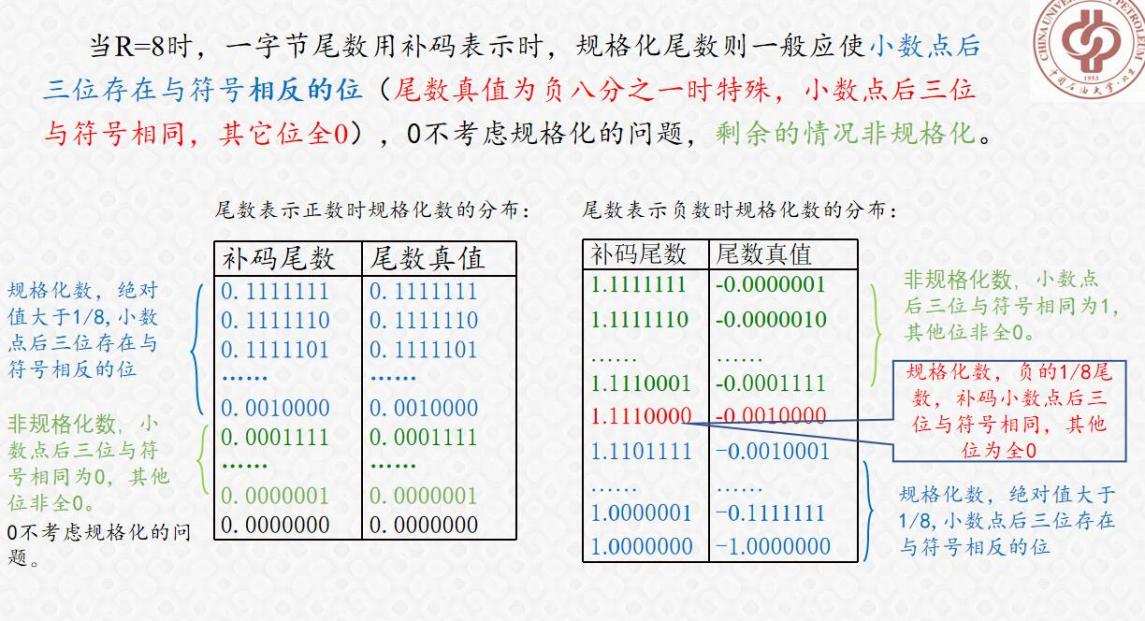
最小负数与最大正数绝对值相同，符号相反。

可得原码E=0,11111和 M=1.1111111，对应的浮点数的真值

-0.1111111111111112×231

= -（1-2-7）×231

1. 当R=8，尾数和码用二进制补码表示时，规格化尾数如何确定小数点后三位的取值？



1. 已知*x*=-101101，y=101111，求它们的移码。

已知*x*=-101101，[-101101]补=1,010011，[-101101]移=0,010011

求负数的移码，由补码过渡

y=101111，[ *y*]移=1,101111

求正数的移码，有效数字前加符号1即可。

1. 已知[*x*]移=1,0101101、[ *y*]移=0,0111101，求*x*和*y*的真值。

解：已知[*x*]移=1,0101101，*x*=0101101

求正数移码的真值，去掉符号1即可。

已知[ *y*]移=0,0111101，[ *y*]补=1,0111101，*y*=-1000011

求负数移码的真值，由补码过渡。

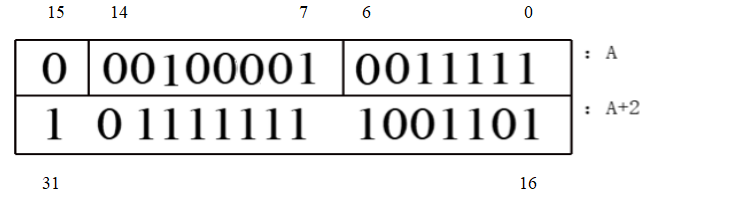
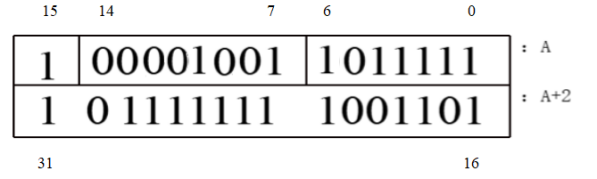
1. 已知[*x*]移=1,0111101、[ *y*]移=0,0111011，求[ *x*+*y*]移。

解：已知[*x*]移=1,0111101、[ *y*]移=0,0111011，则

[*x*]移+[ *y*]移=1,0111101+0,0111011=1,1111000 （1）

（1）变号得： [ *x*+*y*]移=0,1111000

1. VAX机中的四字节F浮点数分别是(a)和(b),它们表示的真值分别是？



(b)

(a)

解：（a）情况，尾数的小数点后按规格化补码插入值，并正确求阶码的真值：

M=0.100111111011111111001101，E=00100001-10000000=-1011111

浮点数真值

0.1001111110111111110011012×2-95

（b）情况，尾数的小数点后按规格化补码插入值，并正确求阶码的真值：

M=1.010111111011111111001101，E=00001001-10000000=-1110111

浮点数真值

-0.1010000001000000001100112×2-119

1. 已知真值x=119.125和y=-235.625，求它们的VAX机四字节F浮点数的表示。

解：

x=119.125=96+23+0.125=64+32+16+7+0.125=(1110111.001)2=(0.1110111001)2×2+7

M=0.111011100100000000000000

E=10000000+111=10000111

VAX机四字节F浮点数的尾数按规格化数省略小数点后最高位：

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 10000111 | 1101110 | ：A |
| 01 0000000  0000000 | | | ：A+2 |

y=-235.625=-128-64-32-11-0.625=-(11101011.101)2=(-0.11101011101)2×2+8

M=1.000101000110000000000000

E=10000000+1000=10001000

VAX机四字节F浮点数的尾数按规格化数省略小数点后最高位：

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 10001000 | 0010100 | ：A |
| 0 1100000  00000000 | | | ：A+2 |

1. 已知十进制数x=125和y=832，用8421BCD码进行加法运算。

解：

0001 0010 0101

+ 1000 0011 0010

1001 0101 0111

x+y=（1001 0101 0111)8421BCD

1. 已知十进制数x=999和y=879，用BCD码进行加法运算。

解：

1001 1001 1001

+ 1000 0111 1001

1 00101 00011 0010

+ 0110 0110 0110

0001 1000 0111 1000

x+y=（0001 1000 0111 1000)BCD

.