习题一

一．说明计算机中浮点数的表示格式和进行四则运算的方法

1.浮点数的表示格式：

计算机中的浮点数通常分为32位单精度和64位双精度两种主要形式。一个浮点数由三个部分组成：符号位、指数部分和尾数（有效数字）部分。

符号位用于表示浮点数的正负，0 表示正数，1 表示负数，单精度浮点数和双精度浮点数的符号数都是 1 位；指数部分用于表示指数值，单精度浮点数有 8 位指数位，采用偏移 127 的方式存储，即存储的实际值是$E_{存储}=E_{实际}+127$，双精度浮点数有 11 位指数位，采用偏移 1023 的方式存储，即存储的实际值是$E_{存储}=E_{实际}+1023$；尾数部分用于表示数的精确度，通常在二进制表示中，尾数会省略掉开头的 1，因为它总是 1，单精度浮点数有 23 位尾数位，双精度浮点数有 52 位尾数位。

2.浮点数进行四则运算的方法：

浮点数的四则运算包括加法、减法、乘法和除法。其步骤分别如下：

加法和减法：

对阶：使两个操作数的指数部分对齐。如果不相等，则将指数较小的那个浮点数的尾数进行右移，直到两个指数相等。

尾数相加或相减：对齐指数后，将尾数相加或相减。

规格化：将结果规格化，使得尾数部分在 1 和 2 之间。

舍入：根据舍入规则，将尾数舍入到合适的精度。

溢出和下溢处理：检测并处理可能出现的溢出（结果过大）或下溢（结果过小）的情况。

乘法：

符号位计算：结果的符号位由两个操作数符号位的异或确定。

指数相加：将两个操作数的指数部分相加，并减去偏移量（对于单精度，减去 127；对于双精度，减去 1023）。

尾数相乘：将两个操作数的尾数部分相乘。

规格化：将结果规格化，使得尾数部分在 1 和 2 之间。

舍入：根据舍入规则，将尾数舍入到合适的精度。

溢出和下溢处理：检测并处理可能出现的溢出或下溢的情况。

除法：

符号位计算：结果的符号位由两个操作数符号位的异或确定。

指数相减：将被除数的指数部分减去除数的指数部分，并加上偏移量（对于单精度，加上 127；对于双精度，加上 1023）。

尾数相除：将被除数的尾数部分除以除数的尾数部分。

规格化：将结果规格化，使得尾数部分在 1 和 2 之间。

舍入：根据舍入规则，将尾数舍入到合适的精度。

溢出和下溢处理：检测并处理可能出现的溢出或下溢的情况。

二．课后习题：1.1、1.6、1.8、1.16、1.23

1.1 完成下列数制转换：

（1）$(1101011)_2=(6B)_{16}$

（2）$(101111.0111)_2=(57.34)_8$

（3）$(67.24)_8=(110111.0101)_2$

（4）$(15C.38)_{16}=(101011100.00111)_2$

（5）$(5436.15)_O=(101100011110.00110100)_B=(B1E.34)_H$

（6）$(552273)_O=(101101010010111011)_B=(2D4BB)_H$

（7）$(BABE)_H=(1011101010111110)_B=(47806)_{10}$

（8）$(DEAD.0AE)_H=(1101111010101101.000010101110)_B=(57005.04248)_{10}$

1.6 完成下列二进制加、减法：

（1）$(110101)_2+(11001)_2=(1001110)_2$

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

（2）$(101110)_2+(100101)_2=(1010011)_2$

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

（3）$(11011101)_2-(1100011)_2=(1111010)_2$

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

（4） $(1110010)_2 - (1101101)_2 = (101)_2$

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

1.8 分别写出下列十进制数的 8 位二进制原码、反码和补码：

（1）+18

原码：00010010；反码：00010010；补码：00010010；

（2）+115

原码：01110011；反码：01110011；补码：01110011；

（3）+79

原码：01001111；反码：01001111；补码：01001111；

（4）-49

原码：10110001；反码：11001110；补码：11001111；

（5）-3

原码：10000011；反码：11111100；补码：11111101；

（6）-100

原码：11100100；反码：10011011；补码：10011100；

1.16 已知下列机器数，写出它们所对应的真值：

（1） $[x_1]_{原} = 11011$

转换成十进制得真值为：-11

（2） $[x_2]_{反} = 11011$

原码为 10100，转换成十进制得真值为：-4

（3）$[x_3]_{补}=11011$

反码为 11010，原码为 10101，转换成十进制得真值为：-5

（4）$[x_5]_{原}=00000$

转换成十进制得真值为：0

（5）$[x_5]_{反}=01111$

正数原码等于反码，转换成十进制得真值为：+15

（6）$[x_6]_{补}=01000$

正数原码等于补码，转换成十进制得真值为：+8

1.23 完成下列数制转换：

（1）$(1010111)_{BCD}=(57)_{10}$

（2）$(100000111001.01110101)_{BCD}=(839.75)_{10}$

（3）$(1011001111001001)_{余3}=(1000000010010110)_{BCD}$

（4）$(752.18)_{10}=(11101010010.00011000)_{BCD}$