

# 整数运算实验报告

3180103772 张溢弛

## 0.实验描述

- 选择原/补/移码中的一种，或自行设计一种合适的方式(一般由组里最帅的人自行设计，俗称：帅码)表示整数；给出其算术运算算法。要求有理论推导论证，并进行算法分析，编程实现。比较补码、原码、移码及帅码制的表示方法与四则算术运算算法，分析各种码制的优缺点。同时分析字位扩展(8-16, 16-32位)、运算溢出、大小比较等的方法。算法推导证明(如果手写则拍照上交) 计算机程序模拟，程序只可用无符号整数类型 `unsigned int`，不可用 `int`。基本要求实现六个函数：

```
1 typedef unsigned int word;
2 word atom(char*); //字符串转换成对应的二进制。
3 *char* mtoa(word); //二进制转换成字符串。
4 word madd(word,word); //二进制所表示数的加法。
5 word msub(word,word); //减法
6 word mmul(word,word); //乘法
7 word mdiv(word,word); //除法
8 word mmmod(word,word); //取余
```

## 1.开发环境

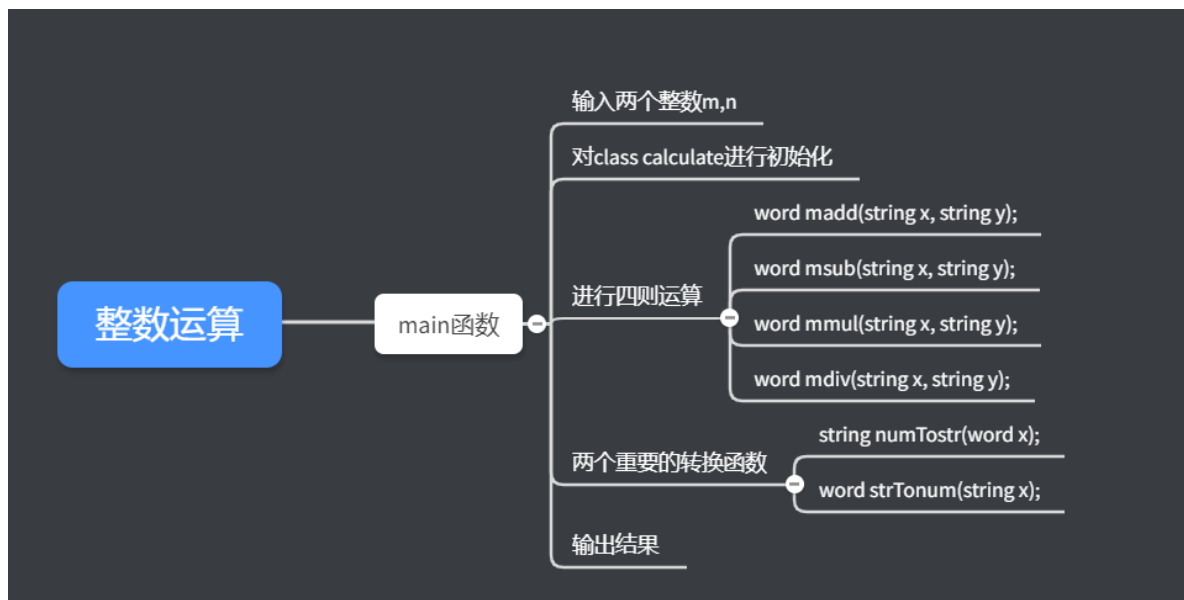
- OS: Windows 10
- IDE: Visual Studio 2017
- Language: C++

## 2.程序结构

- 本程序实现了八位二进制的无符号整数的四则运算，包括加减乘除，对于输入负数的情况，会将负数自动转换成符合条件的八位二进制无符号整数
- 本程序的基本结构如下，核心的类是 `calculate` 类

```
1 class calculate
2 {
3 private:
4     word num1, num2;
5     string bin1, bin2;
6
7 public:
8     calculate(Word a,Word b);
9     //~calculate();
10    string numToStr(word x);
11    word strTonum(string x);
12    word madd(string x, string y);
13    word msub(string x, string y);
14    word mmul(string x, string y);
15    word mdiv(string x, string y);
16    void run();
17
18 };
```

- 程序的结构框图如下



### 3. 算法分析与数据表达

- 本实验中所有数据转化成8位移码来表示
- 加法
  - 采用原码直接相加的算法，设置数组存储进位的情况来应对相加产生进位的情况
  - 由于是采用原码直接相加，因此不需要进行数学意义上的证明
- 减法
  - 采用补码相加的算法，将被减数转换成补码，以达到将加法转换成减法的目的，其中八位无符号数的原码---补码转换规则是
    - 将被减数的每一位二进制表示取反，并加上1，可以将减去被减数转换成加上被减数的补码，在运算结果上是等效的
  - 证明方法如下

$$(X - Y) = (X + (-Y)) = (X + Y_{\text{反}} - 0x\text{FF}) = (X + Y_{\text{补}})$$

- 乘法
  - 模拟计算机内部的运算模式，将乘法转换为若干次的加法，将结果先设定为0，每次将结果与因数相加，另一个因数-1，直到被减的因数变成1为止。而加法采用原码相加的规则
- 除法
  - 模拟计算机内部的运算模式，将除法转换为若干次减法，先设定商为0，再将除数减去除数，直到被除数小于除数为止，每次相减后商增加1，减法的规则采用补码相加法

### 4. 原码，移码，补码的比较/字位扩展和大小比较

- 原码：源码就是符号位加上真值的绝对值，第一位表示符号其他位表示值，其优点有
  - 对数据的表示非常直观，符号位和值分开便于人的阅读
  - 容易判断运算的溢出

但是缺点是

- 0有+0和-0之分
- 比较两个数的大小时需要先比较符号位
- 用原码进行加减运算之前必须判断符号位，否则会出错

- 移码是在原码基础上增加一定的偏移量，N位整数的偏移量通常取 $2^{(N-1)}$ ，其目的是将一个被编码数X转换成一个非负数，其优点有
  - 相对于原码，可以方便地比较两数的大小和进行两数的减法运算
  - 解决了+0和-0的问题
- 缺点为
  - 依然没有解决原码的加减运算不能合并成一种操作的问题
  - 乘法难以实现
- 补码：补码的优点有
  - 补码使得加减法成为同一种操作
  - 补码的使用解决了原码中相反数相加不为0的问题
  - 在字位扩展中有比较好的性能
- 字位扩展
  - 带符号扩展：有符号整数按照符号扩展的形式扩展高位，即最高位是1则高位全部补1，最高位是0全部补0。
  - 无符号扩展：按照最高位直接都补0的方式扩展
- 大小比较
  - 原码和补码：先比较符号位，若符号位不同则可以直接得出比较结果，若符号位相同且都是正数则从高位起一位一位往下比较
  - 移码：类似于源码和补码，但是不需要比较符号位
- 溢出判断
  - 每次运算结束后只取最低的八位作为结果

## 5. 实验结果

- 输入两个整数

```
Welcome to the integer calculator developed by zyc
Please input 2 unsigned integer in a line
15 4
00001111
00000100
The result of add is 19
The result of add in binary is 00010011
The result of sub is 11
The result of sub in binary is 00001011
The result of mul is 60
The result of mul in binary is 00111100
The result of div is 3
The result of div in binary is 00000011
```

- 程序会自动输出两个数的八位二进制无符号形式，并将四则运算的结果分别以十进制和八位二进制的形式输出，经检验结果都是正确的
- 输入的内容中包含负数，结果依然正确

```
Welcome to the integer calculator developed by zyc
Please input 2 unsigned integer in a line
3 -1
00000011
11111111
The result of add is 2
The result of add in binary is 00000010
The result of sub is 4
The result of sub in binary is 00000100
```

## 6. 心得体会

- 这次试验的主要内容是整数运算的模拟和程序实现，我感觉收获很大，对楼老师课堂上讲的计算机系统内部整数四则运算的方式有了更深入的理解。

## 7. 源代码

- calculate.h

```
1 | #include<iostream>
```

```

2  #include<string>
3  #include<vector>
4  #include<stdlib.h>
5  #include<stdio.h>
6  using namespace std;
7
8  typedef int Word;
9  typedef unsigned int word;
10
11 class calculate
12 {
13 private:
14     word num1, num2;
15     string bin1, bin2;
16
17 public:
18     calculate(Word a, Word b);
19     //~calculate();
20     string numToStr(word x);
21     word strTonum(string x);
22     word madd(string x, string y);
23     word msub(string x, string y);
24     word mmul(string x, string y);
25     word mdiv(string x, string y);
26     void run();
27
28 };

```

- calculate.cpp

```

1  #include "calculate.h"
2
3  #include<iostream>
4  #include<string>
5  #include<vector>
6  #include<stdlib.h>
7  #include<stdio.h>
8  using namespace std;
9
10 calculate::calculate(Word a, Word b)
11 {
12     //change to a suitable value for the 8-bit calculate
13     while (a < 0 || b < 0) {
14         if (a < 0)
15             a += 256;
16         if (b < 0)
17             b += 256;
18     }
19
20     if (a > 256) {
21         a = 256;
22     }
23     if (b > 256) {
24         b = 256;
25     }
26 }

```

```

25     }
26     num1 = a;
27     num2 = b;
28     bin1 = numToStr(num1);
29     bin2 = numToStr(num2);
30     cout << bin1 << endl;
31     cout << bin2 << endl;
32 }
33
34 word calculate::strTonum(string x)
35 {
36     int change[8] = { 128,64,32,16,8,4,2,1 };
37     int i,result = 0;
38     for (i = 0; i < x.size(); i++) {
39         if (x[i] == '1')
40             result += change[i];
41     }
42     return result;
43 }
44
45 void calculate::run()
46 {
47
48     cout << "The result of add is " << madd(bin1, bin2) << endl;
49     cout << "The result of add in binary is " <<
numToStr(madd(bin1, bin2)) << endl;
50
51     cout << "The result of sub is " << msub(bin1, bin2) << endl;
52     cout << "The result of sub in binary is " <<
numToStr(msub(bin1, bin2)) << endl;
53
54     cout << "The result of mul is " << mmul(bin1, bin2) << endl;
55     cout << "The result of mul in binary is " <<
numToStr(mmul(bin1, bin2)) << endl;
56
57     cout << "The result of div is " << mdiv(bin1, bin2) << endl;
58     cout << "The result of div in binary is " <<
numToStr(mdiv(bin1, bin2)) << endl;
59 }
60
61 string calculate::numToStr(word x)
62 {
63     vector<int> bin;
64     while (x != 0) {
65         bin.push_back(x % 2);
66         x /= 2;
67     }
68     string result = "";
69     for (int i = 0; i < 8 - bin.size(); i++) {
70         result += '0';
71     }
72     for (int i = bin.size() - 1; i >= 0; i--) {
73         if (bin[i])
74             result += '1';
75         else

```

```

76         result += '0';
77     }
78     return result;
79 }
80
81
82 word calculate::madd(string x, string y)
83 {
84     int i,a,b,bit,cf = 0;
85     vector<char> bin;
86     for (i = 7; i >= 0; i--) {
87         a = x[i] - '0', b = y[i] - '0';
88         bit = a + b + cf;
89         if (bit % 2 == 0) {
90             bin.push_back('0');
91         }
92         else {
93             bin.push_back('1');
94         }
95         if (bit >= 2) {
96             cf = 1;
97         }
98         else {
99             cf = 0;
100         }
101     }
102     string result = "";
103     for (i = 7; i >= 0; i--) {
104         result += bin[i];
105     }
106
107     return strTonum(result);
108 }
109
110
111 word calculate::msub(string x, string y)
112 {
113     int i, a, b, bit, cf = 0;
114     for (i = 0; i < 8; i++) {
115         if (y[i] == '1')
116             y[i] = '0';
117         else
118             y[i] = '1';
119     }
120     y = numTostr(madd(y, "00000001"));
121
122     //cout << y << endl;
123
124     vector<char> bin;
125     for (i = 7; i >= 0; i--) {
126         a = x[i] - '0', b = y[i] - '0';
127         bit = a + b + cf;
128         if (bit % 2 == 0) {
129             bin.push_back('0');
130         }

```

```

131         else {
132             bin.push_back('1');
133         }
134         if (bit >= 2) {
135             cf = 1;
136         }
137         else {
138             cf = 0;
139         }
140     }
141     string result = "";
142     for (i = 7; i >= 0; i--) {
143         result += bin[i];
144     }
145
146     //cout << "The result of sub in binary is " << result << endl;
147     return strTonum(result);
148 }
149
150 word calculate::mmul(string x, string y)
151 {
152     int i, j;
153     string result = "00000000";
154     for (i = 0; i < strTonum(x); i++) {
155         result = numToStr(madd(result, y));
156     }
157     return strTonum(result);
158 }
159
160
161 word calculate::mdiv(string x, string y)
162 {
163     word count = 0;
164     while (strTonum(x) >= strTonum(y)) {
165         x = numToStr(msub(x, y));
166         count++;
167     }
168     return count;
169 }

```

- main.cpp

```

1  #include<iostream>
2  #include<string>
3  #include<vector>
4  #include<stdlib.h>
5  #include<stdio.h>
6  using namespace std;
7
8  typedef unsigned int word;
9  #include"calculate.h"
10
11 int main()
12 {

```

```
13     cout << "Welcome to the integer calculator developed by zyc" <<
endl;
14     cout << "Please input 2 unsigned integer in a line" << endl;
15     Word a, b;
16     cin >> a >> b;
17     calculate x(a, b);
18     x.run();
19     system("pause");
20     return 0;
21 }
```