

Эмуляция троичной арифметики

Гиниятуллин В.М., Миргалиев В.Д.

Уфимский государственный нефтяной технический университет

tnsfloat@mail.ru

В работе обсуждается библиотека программной эмуляции арифметических операций в троично – сбалансированной системе счисления (ТСС).

Введение

Троично – сбалансированная система счисления – это позиционная система, алфавит которой состоит из трех символов -1, 0, +1. Одна из первых, промышленно выпускавшихся, ЭВМ – «Сетунь» [1] была основана именно на этой системе счисления.

В настоящее время отсутствие троичной элементной базы не позволяет аппаратно реализовать преимущества ТСС, поэтому была написана библиотека программной эмуляции арифметических операций ТСС – чисел с плавающей запятой.

Описание библиотеки

ТСС – числа эмулируются с помощью структуры TNSFLOAT, в которой содержится 2-а статических массива типа signed char для хранения экспоненты и мантииссы. Каждый элемент этих массивов – это 1-н разряд ТСС – числа (трит) и может содержать значения -1, 0 +1. Кроме того, в структуре объявлены: конструктор (обнуляет экспоненту и мантииссу), перегруженный оператор присвоения (преобразует число типа double в тип TNSFLOAT) и перегруженный оператор присвоения (копирует входной параметр типа TNSFLOAT).

Вне пределов структуры TNSFLOAT определен набор операторов (+, -, *, /), операции больше/меньше, функция abs (абсолютное значение), функция tnsToFloat (преобразует TNSFLOAT число в double) и функция printTnsFloat (выводит TNSFLOAT число). Ниже приведен результат умножения нескольких чисел в десятичном и TNSFLOAT виде.

13.200 * 0.700 = 9.24000 0.100171777 t000010

12.200 * 1.700 = 20.74000 0.171071700 t000011

11.200 * 2.700 = 30.24000 0.101017177 t000011

10.200 * 3.700 = 37.74000 0.111771700 t000011

9.200 * 4.700 = 43.24000 0.177711701 t000177

8.200 * 5.700 = 46.74000 0.177177170 t000177

7.200 * 6.700 = 48.24000 0.177101701 t000177

6.200 * 7.700 = 47.74000 0.177107170 t000177

5.200 * 8.700 = 45.24000 0.177001701 t000177

4.200 * 9.700 = 40.74000 0.177777170 t000177

$3.200 * 10.700 = 34.24000 \ 0.117117011 \ t000011$
 $2.200 * 11.700 = 25.74000 \ 0.100771707 \ t000011$
 $1.200 * 12.700 = 15.24000 \ 0.177017011 \ t000011$
 $0.200 * 13.700 = 2.74000 \ 0.107170000 \ t000017$
 $-0.800 * 14.700 = -11.76000 \ 0.770170110 \ t000010$
 $-1.800 * 15.700 = -28.26000 \ 0.700771700 \ t000011$
 $-2.800 * 16.700 = -46.76000 \ 0.711711701 \ t000177$
 $-3.800 * 17.700 = -67.26000 \ 0.717777170 \ t000177$
 $-4.800 * 18.700 = -89.76000 \ 0.707001710 \ t000177$

Нормализованная мантисса TNSFLOAT – числа в старшем разряде содержит 0, вместо символа -1 используется символ 7. Экспонента отделена буквой t и содержит целое TNSFLOAT – число, ведущие нули не значимы.

Для реализации арифметических операций использовались классические алгоритмы «в столбик» [2].

Использование библиотеки

Операции сложения, умножения, деления и инверсии (изменение знака) реализованные как экспортируемые функции. Операция вычитания реализована через инверсию и сложение. Для использования библиотеки необходимо объявить 4-е указателя на функцию и подключить динамически подключаемую библиотеку tnsFloatDLL. В приложении имеется примеры, написанный в среде QtCreator (v. 2010.05).

Сравнение TNSFLOAT и вещественных двоичных чисел

Для сравнения с TNSFLOAT – числами на языке ассемблера был разработан ещё один тип данных floatSSE, что гарантирует использование 32-х битных двоичных чисел с плавающей запятой (тип float) [3].

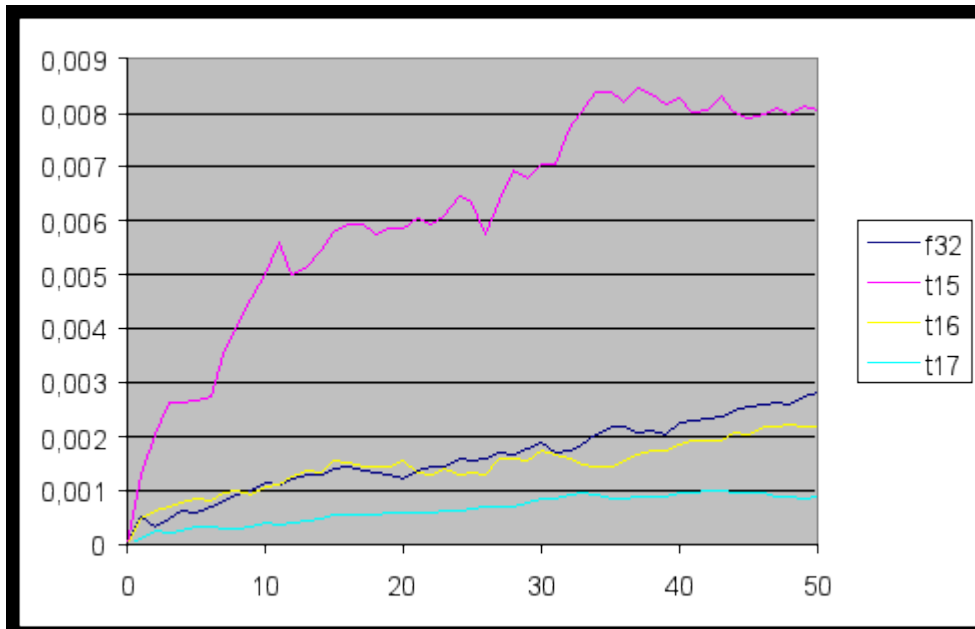
Оба типа данных тестировались на примере обращения квадратных матриц. Матрицы, размером от 4x4 до 14x14, случайным образом заполнены целыми числами. Приложение читает исходную матрицу из файла, затем в цикле дважды обращает её, сравнивает с исходной матрицей и выводит накопленную ошибку округления. Каждое последующее двойное обращение увеличивает ошибку округления.

С вещественными 32-х битными двоичными числами сравнивались TNSFLOAT – числа с экспонентой в 5 трит и мантиссой в 15, 16 и 17 трит. Мантисса float числа содержит 23 бита без ведущей единицы, т.е. её мощность $2^{24} = 16\ 777\ 216$, мощность мантиссы TNSFLOAT – числа $(3^n - 1)/2$:

$$(3^{15} - 1)/2 = \pm 7\ 174\ 453$$

$$(3^{16} - 1)/2 = \pm 21\,523\,360$$

$$(3^{17} - 1)/2 = \pm 64\,570\,081.$$



На рисунке представлен результат вычислительного эксперимента для матрицы 9x9 для 50-и двойных обращений, имеются 4-е графика зависимости ошибки округления от количества двойных обращений. Видно, что с увеличением мощности мантиссы ошибки округления уменьшаются. Кроме того мощности 16-и разрядного TNSFLOAT – числа и float – числа близки, сопоставимы и величины накопленных ошибок.

Для других матриц получены аналогичные результаты.

Заключение

Экспериментально подтверждена большая экономичность троично – сбалансированной системы счисления, по сравнению с двоичной системой счисления. Сопоставимые результаты получены для 16-и разрядной мантиссы TNSFLOAT – числа, что на 7-мь разрядов меньше чем в float – числах.

С помощью созданной библиотеки можно реализовывать вычисления произвольной разрядности.

К работе приложены библиотеки: tnsFloatDll.dll и arithmeticSSE.dll, пример их использования в среде QtCreator (v. 2010.05) и результаты вычислительного эксперимента для квадратных матриц, размерами от 4x4 до 14x14.

Литература

1.
Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина "Сетунь". - М.: Изд-во Моск. ун-та, 1965.
2.
Альварес Х.Р. Алгоритмы деления и извлечения квадратного корня в троичной симметричной системе. – М.: Вестн. Моск. ун-та. Сер. 15. Вычислительная математика и кибернетика. 2008. № 2. С. 42-45.
3.
Зубков С.В. Assembler для DOS, Windows и UNIX 2-е издание. Язык низкого уровня М: ДМК 2000 г. 608 с.