

ЭФФЕКТ КОМПЕНСАЦИИ ОШИБОК ОКРУГЛЕНИЯ В ТРОИЧНО-СБАЛАНСИРОВАННОЙ СИСТЕМЕ СЧИСЛЕНИЯ

В. М. Гиниятуллин [✉], М. А. Салихова

Уфимский государственный нефтяной технический университет, Уфа, Россия

[✉]E-mail: fentazer@mail.ru

Рассмотрена реализация троичной сбалансированной системы счисления (далее – ТСС) для дробных чисел в экспоненциальном формате. Даны графические представления дробей с избытком/недостатком. В программную модель троичного сопроцессора добавлен регистр-накопитель ошибок округления. Сформулирован алгоритм работы регистра-накопителя. Для кодирования символа -1 используется символ 7. Приведены примеры некоторых чисел в виде дробей, в десятичном представлении, в ТСС-виде с избытком и недостатком. Подробно рассмотрена процедура численного интегрирования методом прямоугольников для значений функции в ТСС-виде с недостатком. За 6 операций сложения в регистре-накопителе дважды фиксируется значение, компенсирующее ошибку. Окончательный результат совпадает с ожидаемым с точностью до последнего знака. Во втором числовом примере разбирается операция скалярного умножения. Операнды представлены в ТСС-виде с избытком. Анализ значений промежуточных результатов показал, что ошибка округления накапливается, однако вследствие компенсации точность конечного результата возвращается к последнему значащему разряду. В последнем примере демонстрируется использование иррациональных дробей, но, несмотря на это, окончательный результат совпадает с точным – получено целое число. Сделан вывод о том, что реализация регистра-накопителя возможна только в троичной сбалансированной системе счисления. Поставлена цель в дальнейшем определить минимально достаточное количество разрядов троичного вычислителя для задач автоматического управления. Кроме того, планируется аппаратная реализация в микросхемах программируемых логических интегральных схем троичного сопроцессора с компенсацией ошибок в идеологии single instruction multiple data.

Ключевые слова: троичная сбалансированная система счисления, экспоненциальный формат числа, арифметический сопроцессор, вычислительный регистр, регистр/накопитель, представление дробей с избытком/недостатком, компенсация ошибок округления.

EFFECT OF ROUNDING ERROR COMPENSATION IN THE TERNARY BALANCED NUMBER SYSTEM

V. M. Giniyatullin [✉], M. A. Salikhova

Ufa State Petroleum Technological University, Ufa, Russia

[✉]E-mail: fentazer@mail.ru

The implementation of the ternary balanced number system (TBNS) for fractional numbers in scientific notation is considered. Graphical representations of fractions with excess/deficiency are given. In the program model of the ternary coprocessor, the accumulator register of rounding errors is added. The algorithm of the accumulator register operation is formulated. The character 7 is used for encoding -1 . Examples of some numbers are given in the form of fractions, in decimal representation, and in the TBNS form with excess and deficiency. The authors consider in detail the procedure of numerical integration for function values in the TBNS form with a deficiency by the rectangle method. For six addition operations, the error-compensating value is accumulated twice in the storage register. The final result is the same as the expected one, up to the last digit. In the second

numerical example, the scalar multiplication is analyzed. Operands are represented in the TBNS form with deficiency. Analysis of the values of intermediate results showed that the rounding error accumulates, but due to compensation, the accuracy of the final result returns to the last significant digit. The last example demonstrates the use of irrational fractions, although the final result is the same as the exact one, an integer is obtained. It is concluded that the implementation of storage registers is possible only in the balanced ternary. The object in view is to determine the minimum sufficient number of digits of the ternary calculator for automatic control problems. In addition, it is planned to implement hardware in programmable logic integrated circuits of a ternary coprocessor with error compensation in the ideology of single instruction multiple data (SIMD).

Keywords: balanced ternary, ternary number system, scientific notation, math coprocessor, computational register, accumulator register, representation of fractions with excess/deficiency, rounding error compensation.

Введение

Троичная сбалансированная система счисления – это позиционная система счисления, которая содержит в своем алфавите 3 символа: -1 ; 0 ; $+1$. Знаки перед единицами «встроены» в свои символы и иногда разряды ТСС-чисел так и именуются – «минус/плюс». Для обозначения -1 можно использовать любой символ кроме 0 и 1 , в данной работе используется символ 7 .

Впервые ТСС была аппаратно реализована Н. П. Брусенцовым в ЭВМ «Сетунь» [1], позднее – в California Polytechnic State University [2], в наше время также находятся энтузиасты-электронщики [3]. Работы [4–5] больше относятся к области программной эмуляции троичности, поскольку при необходимости использование программируемых логических интегральных схем (далее – ПЛИС) позволяет аппаратно реализовывать любые алгоритмы [6].

Основной причиной невостребованности ТСС является отсутствие преимуществ перед двоичной системой счисления. На нескольких примерах разной степени искусственности продемонстрируем одно из преимуществ ТСС, а именно эффект компенсации ошибок округления. В работе [1, с. 2] утверждается, что «...наилучшее округление числа до k верных троичных знаков получается отбрасыванием младших знаков, начиная с $(k+1)$ -го, что избавляет от необходимости устраивать в машине аппарат округления...», но для того, чтобы вычисления в ТСС не накапливали ошибок, необходимо предпринимать специальные действия.

Дробные числа

Дробные числа в ТСС реализуются стандартным методом как отрицательные степени тройки. Аналогично другим системам счисления в ТСС имеются бесконечные дроби: иррациональные и периодические. Особенностью ТСС является представление дроби $1/2$ – это бесконечная периодическая дробь, но у нее есть два вида представления $0,(1)$ и $1,(7)$. Первый вариант вычисляется следующим образом:

$$x = \frac{1}{3} + \frac{1}{9} + \frac{1}{27} + \frac{1}{81} + \dots \quad (1)$$

Сумма (1) будет приближаться к $1/2$ с недостатком. Второй вариант:

$$x = 1 - \frac{1}{3} - \frac{1}{9} - \frac{1}{27} - \frac{1}{81} - \dots \quad (2)$$

Разница (2) будет приближаться к $1/2$ с избытком. На рисунке по оси ординат отложено количество степеней тройки/разрядов числа, по оси абсцисс – соответствующее значение числа. Видно, что $1/2$ – это асимптота, к которой стремится значение числа с ограниченным количеством разрядов сверху (избыток) или снизу (недостаток). Третья линия – это представление числа $1/4$, ТСС-вид – $0,(17)$:

$$x = \frac{1}{3} - \frac{1}{9} + \frac{1}{27} - \frac{1}{81} + \dots \quad (3)$$

Выражение (3) можно понимать как знакопеременный ряд, затухающий вокруг некоторого значения.

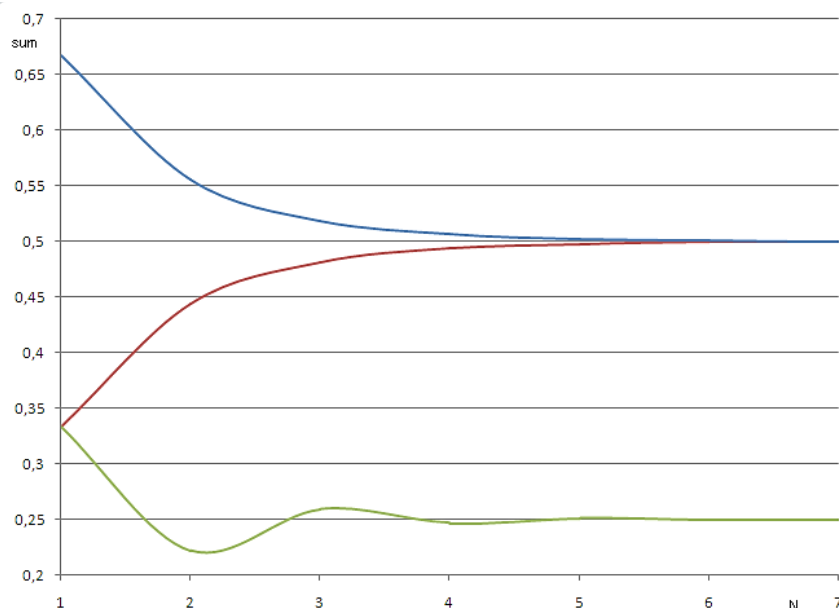


Рисунок. Графическое представление дробей 1/2 и 1/4
Примечание: составлено авторами.

В таблице 1 даны представления некоторых ТСС-дробей в экспоненциальном виде. Первые 6 чисел – это 1/2, помноженная на некоторую степень тройки, в представлениях с недостатком/избытком. Длина периода дроби 1/4 – 2 троичных разряда. На практике количество разрядов ограничивают, следовательно, состояние недостаток/избыток периодических дробей определяется знаком младшего разряда. Длина периода дроби 6/7 – 6 десятичных и 7 троичных разрядов. Корень из 7 – это иррациональное число, поэтому значения в соответствующей строке табл. 1 равны друг другу примерно. Целые числа в экспоненциальном ТСС-виде могут быть представлены точно, если количество их разрядов меньше длины мантииссы.

Таблица 1

Примеры ТСС-чисел

Десятичные	Дроби	Недостаток	Избыток
4,5	9/2	$1,(1) \times 3^1$	$1,(7) \times 3^2$
1,5	3/2	$1,(1) \times 3^0$	$1,(7) \times 3^1$
0,5	1/2	$1,(1) \times 3^{-1}$	$1,(7) \times 3^0$
0,1(6)	1/6	$1,(1) \times 3^{-2}$	$1,(7) \times 3^{-1}$
0,0(5)	1/18	$1,(1) \times 3^{-3}$	$1,(7) \times 3^{-2}$
0,0(185)	1/54	$1,(1) \times 3^{-4}$	$1,(7) \times 3^{-3}$
0,25	1/4	$1,(71) \times 3^{-1}$	
0,(857142)	6/7	$1,(0770011) \times 3^0$	
2,64575131	$\sqrt{7}$	$1,070711 \times 3^1$	
2,0		$1,7 \times 3^1$	
6,0		$1,7 \times 3^2$	
14,0		$1,777 \times 3^3$	

Примечание: составлено авторами.

Традиционно в программную модель процессоров включают вычислительные регистры и регистры состояния/управления [7], добавим к ним регистр-накопитель ошибок округления. В набор команд обсуждаемого сопроцессора необходимо включить команду сброса/обнуления накопителя. Вычислительные регистры этого сопроцессора это n-разрядные числа с плаваю-

щей запятой, в старшем разряде нуля быть не должно. Уточним, нуль в старшем разряде возможен как промежуточное состояние в процессе выполнения арифметической операции или как результат вычисления, требующий коррекции. До запятой имеется 2 разряда, если старший из них не нуль, то этот результат тоже требует коррекции. Регистр-накопитель – это 4-разрядные числа с фиксированной запятой, нуль в старшем разряде допустим. Как только в старшем разряде накопителя появляется знак, то он изымается и прибавляется к младшему разряду вычислительного регистра.

Метод прямоугольников

Методом прямоугольников на интервале $\{0; 5\}$ с шагом 1 вручную проинтегрируем функцию:

$$y = 4,5 \times \exp(-1,09 \times x). \quad (4)$$

Интеграл функции (4) можно найти аналитически ($\approx 4,11$ [8]), однако эта цифра в данной работе интереса не представляет. Функция (4) получена стандартной аппроксимацией первых шести чисел из табл. 1, операнды текущего и последующих примеров специально подобраны в демонстрационных целях.

В строках 1, 2, 3 табл. 2 производится сложение первых 2 чисел из табл. 1, в столбце 2 – экспоненциальный вид ТСС-чисел в формате пяти разрядов после запятой. Ограничение разрядности дает точность представления $3^{-5} \approx 0,004$. При сложении чисел в экспоненциальном виде необходимо выравнивать степени (совместить запятые), что достигается сдвигом мантиссы (строка 2, столбец 2). Обычно сдвигают вправо меньший по значению операнд, при этом до запятой появляется нуль, а младший разряд усекается (зачеркнутый плюс в строке 2, столбец 2). В 4-м столбце демонстрируется текущее значение регистра-накопителя. Первые три усекаемых разряда перемещаются в накопитель сразу после запятой (строка 2, столбец 4), перед началом вычислений накопитель надо обнулить. Результат первого сложения содержит 2 значащих разряда перед запятой, поэтому необходима коррекция – сдвиг вправо (зачеркнутый минус в строке 3, столбец 2). В 3-м столбце содержится десятичное представление результата после коррекции, в данном случае получено 6,0 – это точное значение (строка 3, столбец 3). К имеющемуся в накопителе плюсу прибавляется усекаемый минус, поэтому в накопителе – 0,000 (строка 4, столбец 4).

Таблица 2

Накопление суммы

	Операнды	ТСС-числа	Десятичные	Накопитель
1	4,5	$1,11111 \times 3^1$		
2	1,5	$0,111114 \times 3^1$		0,100
3	6,0	$17,00007 \times 3^1$	6,0	0,700
4				0,000
5	6,0	$1,70000 \times 3^2$		0,000
6	0,5	$0,00111444 \times 3^2$		0,111
7	6,5	$1,70111 \times 3^2$	6,(481)	0,111
8	6,5	$1,70111 \times 3^2$		0,111
9	0,1(6)	$0,000114444 \times 3^2$		0,111
10	6,(6)	$1,71707 \times 3^2$		1,007
11		$1,71700 \times 3^2$	6,(6)	0,007
12	6,(6)	$1,71700 \times 3^2$		0,007
13	0,0(5)	$0,0000144444 \times 3^2$		0,111
14	6,7(2)	$1,71701 \times 3^2$	6,(703)	0,110
15	6,7(2)	$1,71701 \times 3^2$		0,110
16	0,0(185)	$0,00000444444 \times 3^2$		0,111
17	6,(740)	$1,71701 \times 3^2$		1,071
18		$1,71717 \times 3^2$	6,(740)	0,071

Примечание: составлено авторами.

В строках 5, 6, 7 следующее сложение: 3-е число из таблицы 1 и предыдущий результат. Разница мантисс составляет 3 разряда (строка 6, столбец 2), в регистр-накопитель попадает 0,111 (строка 6, столбец 4), недостаток результата в десятичном виде – 0,019 (строка 7, столбец 3). Круглые скобки в 1-м и 3-м столбцах символизируют периодическую десятичную дробь, которая получается в результате конвертации из ТСС.

В очередном сложении меньший по значению операнд (строки 8, 9, 10 и 11) надо сдвинуть на 4 разряда (строка 9, столбец 2), самая младшая единица усекается, и в накопитель передаются 3 плюса. Суммируя их с содержимым накопителя, в старшем разряде получим плюс (строка 10, столбец 4, курсив). Эта единица прибавляется к младшему разряду вычислительного регистра, его текущее значение – минус (строка 10, столбец 2). В строке 11 в столбце 2 подчеркнутым курсивом обозначен результат компенсации ошибки округления, в накопителе остается 0,007 (строка 11, столбец 4). Таким образом, результат операции после компенсации ошибки совпадает с ожидаемым (строка 11, столбец 3) с точностью до последнего знака.

В следующем сложении меньший по значению операнд (строки 12, 13 и 14) надо сдвинуть на 5 разрядов (строка 13, столбец 2), результат отличается от желаемого примерно на 0,019 с недостатком.

При последнем сложении меньший по значению операнд (строки 15, 16, 17 и 18) сдвигается на все 6 значащих разрядов (строка 16, столбец 2), 3 из них усекаются, 3 складываются с накопителем. В старшем разряде накопителя появляется плюс (строка 17, столбец 4, курсив), его прибавляют к младшему разряду вычислительного регистра (строка 17, столбец 2, курсив). Результат компенсации ошибки округления (строка 18, столбец 2, подчеркнутый курсив) в десятичном виде совпадает с ожидаемым (строка 18, столбец 3).

Скалярное умножение

В предыдущем примере использовалось представление операндов с недостатком. Покажем, что использование представления с избытком обладает аналогичными свойствами. Вычислительная процедура:

$$9/2 \times 1/18 + 3/2 \times 1/16 + 1 \times 1/2 = 1 \quad (5)$$

есть числовой пример скалярного умножения векторов. В таблице 3 этот пример рассмотрен пошагово. Операнды даны в формате 6 знаков после запятой, с избытком, что дает точность представления в десятичной системе $3^{-6} \approx 0,0014$.

Таблица 3

Комбинация умножения и сложения

	Операция	Результат	Накопитель
1	$9/2 \times 1/18 = 1/4$	$1,777777 \times 3^2$	0,171
2		$1,777777 \times 3^{-2}$	
3		$0,171710747474 \times 3^0$	
4		$1,717107 \times 3^{-1} \approx 0,24554$	
5	$3/2 \times 1/6 = 1/4$	$1,777777 \times 3^1$	0,171
6		$1,777777 \times 3^{-1}$	
7		$0,171710747474 \times 3^0$	
8			
9		$1,717100 \times 3^{-1} \approx 0,25103$	
10	$1/4 + 1/4 = 1/2$	$1,717107 \times 3^{-1}$	0,771
11		$1,717100 \times 3^{-1}$	
12		$17,777707 \times 3^{-1}$	
13			
14		$1,777777 \times 3^0 \approx 0,50007$	
15	$1/2 + 1/2 = 1$	$1,777777 \times 3^0$	
16		$1,777777 \times 3^0$	
17		$1,000001 \times 3^0 \approx 1,00137$	

Примечание: составлено авторами.

При умножении чисел в экспоненциальном виде степени складываются, а количество разрядов результата удваивается (строка 3, столбец 2). После усечения и нормализации в накопителе остается троичное значение 0,171 (строка 3, столбец 3). Операнды второго умножения в ТСС-виде отличаются от предыдущего только степенями (строки 5, 6, столбец 2). За счет возврата единицы из накопителя (строка 8, столбец 3, курсив) результат получается с избытком (строка 9, столбец 2, курсив), примерно 0,0013 в десятичной системе. Вычисления скалярного умножения по формуле (5) реализованы последовательно, поэтому ошибка округления первого умножения влияет на результат второго.

После сложения результатов умножений необходимо провести компенсацию ошибки округления (строка 13, столбец 3, курсив), результат получается с избытком (строка 14, столбец 2, курсив), примерно 0,007 в десятичной системе.

Результирующее сложение (строки 15, 16 и 17) приводит к значению, точность которого есть последний значащий троичный разряд (строка 17, столбец 2, курсив). Вернувшись вверх по столбцу 2 можно увидеть, что точность промежуточных результатов может ухудшаться (строка 14, столбец 2), неточным становится 4-й троичный разряд после запятой.

Очевидно, что эффект компенсации ошибок округления (в произвольном порядке следования избытков/недостатков) будет более выраженным.

Метод «туда и обратно»

В вычислительной процедуре

$$(6/7 - 2) \times \sqrt{7} \times \sqrt{7} + 14 = 6 \quad (6)$$

последовательность операций и значения операндов подобраны таким образом, чтобы результат оставался целым числом, практического значения они не имеют и приведены в демонстрационных целях. Можно придумать произвольное количество процедур вида (6), их реализация на обычных калькуляторах неизбежно приводит к появлению ошибок округления. Очевидная причина этого в том, что корни из большинства целых чисел – иррациональные дроби.

В табл. 4 процедура (6) реализована в ТСС-виде. В строках 1, 2 и 3 производится вычитание, сдвиг меньшего аргумента увеличивает накопитель. Нормализация после вычитания реализуется сдвигом влево. При использовании регистра накопителя логично будет заполнить младший разряд вычислительного регистра (строка 4, столбец 2, курсив) содержанием разряда после запятой накопителя (строка 1, столбец 3, курсив). После этого накопитель обнуляется. В результате первого умножения (строки 5, 6 и 7) в накопитель попадает положительное число (строка 7, столбец 3).

Таблица 4

Использование иррациональных дробей

	Операция	Результат	Накопитель
1	6/7-2 = -1,(142857)	$0,1077014 \times 3^1$	0,100
2		$1,700000 \times 3^1$	
3		$0,707701 \times 3^1$	
4		$7,077011 \times 3^0$	
5	$-1,(142857) \times \sqrt{7} \approx -3,023715$	$7,077011 \times 3^0$	0,000
6		$1,070711 \times 3^1$	
7		$7,000710471171 \times 3^1$	
8	$-3,023715 \times \sqrt{7} \approx -8,013468$	$7,000710 \times 3^1$	0,171
9		$1,070711 \times 3^1$	
10		$7,01000070101 \times 3^2$	
11			
12	$-8,013468 + 14 \approx 6$	$0,7010000 \times 3^3$	0,070
13		$1,777000 \times 3^3$	
14		$0,170000 \times 3^3$	
15		$1,700000 \times 3^2$	

Примечание: составлено авторами.

В результате второго умножения в накопитель отсылается отрицательное число (строка 10, столбец 3), суммарно значащим в накопителе остается самый младший разряд. Сдвиг меньшего значения перед сложением прибавляет к накопителю нуль (строка 12, столбец 2). Для нормализации после сложения из накопителя изымается нуль (строка 11, столбец 3, курсив). Окончательное значение совпадает с точным решением, поскольку накопленная ошибка округления находится за пределами разрядности мантиссы.

Выводы

Использование регистра-накопителя при вычислениях в ТСС вполне оправдано. В некоторых случаях (метод прямоугольников и метод «туда и обратно») полученный результат совпадает с точным решением. При применении метода скалярного умножения результирующее значение остается в пределах точности представления дробного числа.

Очевидно, что реализация регистра-накопителя в смещенных системах счисления (двоичной, десятичной и т. п.) бессмысленна, т. к. эффект компенсации ошибки округления в этих системах отсутствует.

Перспективы

В ближайшее время планируется реализовать троичный сопроцессор с компенсацией ошибок в микросхемах ПЛИС, в качестве вычислительной процедуры – воспользоваться какой-либо классической математической моделью автоматического управления (обратный маятник). Цель – определение минимально достаточного количества троичных разрядов в мантиссе вычислительного регистра.

Другой интересной задачей является разработка аппаратной реализации троичного сопроцессора в идеологии SIMD. Такие решения востребованы в прикладных задачах искусственного интеллекта (глубокое обучение), в «тяжелых» математических операциях (обращение матриц) и в некоторых численных методах (методы конечных элементов).

Литература

1. Брусенцов Н. П., Жоголев Е. А., Веригин В. В., Маслов С. П., Тишулина А. М. Малая цифровая вычислительная машина «Сетунь». М. : Изд-во МГУ, 1965. 145 с.
2. Connolly J., Patel C., Chavez A., Nico P. Ternary Computing Testbed 3-Trit Computer Architecture // Computer Engineering Department. California Polytechnic State University of San Luis Obispo, 2008. 192 p.
3. Троичный компьютер своими руками. URL: <https://habr.com/ru/post/496366> (дата обращения: 19.10.2020).
4. Гиниятуллин В. М., Салихова М. А., Чурилов Д. А. Троичная логика. Базис, совершенная форма, правила минимизации // Перспективы науки. 2019. № 12 (123). С. 28–32.
5. Гиниятуллин В. М., Габитов Р. Н. Обработка исключительных ситуаций при арифметических операциях над числами с плавающей запятой в троично сбалансированной системе счисления // Информационные технологии. Проблемы и решения : мат-лы Междунар. науч.-практ. конф. Уфа. 2014. № 1–1 (1). С. 163–166.
6. Селезнева С. Н. Сложность систем функций алгебры логики и систем функций трехзначной логики в классах поляризованных полиномиальных форм // Дискретная математика. 2015. № 27. С. 111–122.
7. Зубков С. В. Assembler для DOS, WINDOWS и UNIX. М. : ДМК, 2000. 608 с.
8. Калькулятор интегралов. URL: www.integral-calculator.ru (дата обращения: 19.10.2020).