**Х.Рамиль Альварес**

# Троичный алгоритм извлечения квадратного корня

В работе [1] приводятся правила и пример извлечения квадратного корня «в столбик» для десятичной системы, после чего формулируются правила для троичной симметричной системы. Для обозначения цифр в троичной симметричной системе счисления будем использовать: “–”, “0” и “+” для цифр, соотвественно, –1, 0 и 1. Шаг основной части алгоритма, т.е. определение очередного трита, состоит из следующих действий:

– текущее значение корня умножается на 2 (+–): сдвиг влево и добавить инвертированное начальное значение;

– если текущее значение подкоренного числа отрицательно, то следующая цифра корня будет “–” или “0”; если положительно, то цифра будет “+” или “0”. Используется ненулевая цифра если модуль результата меньше модуля подкоренного значения, в противном случае цифра равна нулю.

Далее приводится следующий пример извлечения квадратного корня из +−+0+ = 64

| | | |
|---|---|---|
| Подкоренное число: | + −+ 0+ = 64 | |
| + | | |
| + | – | Корень: +0– = 8 |
| | 0−+ | < – берем следующую пару цифр |
| +−b | | < – удвоенный корень, сдвиг плюс b |
| +−− | | < – текущее значение отрицательно b = – |
| −++ | | < – умножаем на b |
| +−− | +−− | < – складываем |
| | 0+0 | < – больше по модулю чем –+, поэтому b = 0 |
| | −+0+ | < – берем еще следующую пару цифр |

| | | |
|---|---|---|
| +−0b | | < – удвоенный корень, сдвиг плюс b |
| +−0− | | < – текущее значение отрицательно b = – |
| −+0+ | | < – умножаем на b |
| +−0− | +−0− | < – складываем |
| | 0000 | |

Заметим, что в приведенном примере крайне левая пара содержит лишь один трит и поэтому значение первого трита корня однозначно определятся (“+”). В случае, если первая пара подкоренного числа содержит два трита, ей может соответствать один трит (“+”) или пара тритов (+–). Кроме того, в примере не рассмотрены важные детали алгоритма, когда необходимо оценивать необработанную часть подкоренного числа.

В работе [2] описан алгоритм вычисления квадратного корня для нормализованных троичных чисел, определенных в [3], т.е. равных нулю или принадлежащих интервалу (0.5 ; 1.5). Описание алгоритма дано на языке Алгол, т.е. не привязано к конкретной системе счисления. К сожалению, в описании допущена ошибка для случая, когда для определения значения очередного трита необходимо оценивать необработанную часть подкоренного числа.

В данной статье описывается алгоритм вычисления квадратного корня «в столбик» для троичной симметричной системы в общем случае, т.е. без ограничений, сделанных в работе [2].

## Общие правила извлечения квадратного корня

Также как при вычислении квадратного корня для систем с неотрицательными значениями цифр подкоренное число N разбивается на пары тритов влево и вправо от точки. Обозначим через $N_i$ – целое число, получаемое из первых слева направо i пар тритов, $A_i$ – целое значение корня из этого числа, $D_i$ – i–я пара тритов, $d_i$ – значение i–го трита корня.

Алгоритм извлечения квадратного корня в "столбик" для троичной системы счисления можно описать следующими формулами:

$$N_1 = D_1,$$
$$A_1 = d_1,$$
$$B_1 = N_1 – A_1^2,$$
$$N_i = A_i^2 + B_i,$$
$$C_{i+1} = B_i \bullet 3^2 + D_{i+1}\ ,$$
$$B_{i+1} = C_{i+1} – 2 \bullet A_i \bullet 3 \bullet d_{i+1} – d_{i+1}^2,$$
$$A_{i+1} = A_i \bullet 3 + d_{i+1}.$$

Отметим, что действия производятся над целыми числами, а точка в результате будет после k–го трита, если в подкоренном числе она стояла после k–й пары.

Заметим, что $B_i \bullet 3^2 + D_{i+1}$ при ручном использовании алгоритма выполняется приписыванием справа к значению $B_i$ двух тритов пары $D_{i+1}$, а $2 \bullet A_i \bullet 3 + d_{i+1}$ и $A_i \bullet 3 + d_{i+1}$ – приписыванием трита $d_{i+1}$ к числам соответственно $2 \bullet A_i$ и $A_i$.

**Определение $d_1$**

Если после разбиения числа на пары тритов рассмотреть подкоренное число как число с точкой после первой пары, то получим значения в интервале (.5:4.5). Корень из .5 больше чем .5, т.е первый трит корня есть "+". Корень из 4.5 больше 2, т.е. начальное значения корня должно содержать два трита "+–". Поэтому следует рассматривать два случая:

– первая пара состоит из одного трита и имеем число в интервале (.5:1.5). Корень будет, грубая оценка, больше .7 и меньше 1.23, т.е. первый трит корня $d_1$ будет "+";

– первая пара содержит два трита и имеем число в интервале (1.5 : 4.5). Корень будет, грубая оценка, больше 1.22 и меньше 2.24, т.е. первой паре может соответствовать один трит корня "+" или два трита "+–". Если добавить слева нулевой трит, то получим число в интервале (.166… :.5) и первый трит корня будет "0" или "+". Если подкоренное число больше ¼, корень больше чем ½, т.е. первый трит корня $d_1$ будет "+". В противном случае первый трит корня $d_1$ будет "0". Заметим, что ¼ не представляется точно в троичной симметричной системе и имеет представление в виде бесконечной последовательности **.+–+–+–+–** . . .

**Определение $d_{i+1}$**

При $d_{i+1}$ отличном от "0" имеем

$$B_{i+1} = (C_{i+1} - 1) - 2 \bullet A_i \bullet 3 \bullet d_{i+1},$$

т.е. $d_{i+1}$ есть частное от деления $C_{i+1}-1$ на $2 \bullet A_i \bullet 3$ и вследствие алгоритма деления в троичной системе $d_{i+1}$ равно единице, если

$$2 \bullet |C_{i+1} - 1| \geq 2 \bullet A_i \bullet 3,$$

или проще

$$|C_{i+1} - 1| > A_i \bullet 3.$$

При $C_{i+1} > 0$ если $C_{i+1} > A_i \bullet 3$ и $B_{i+1} = C_{i+1} - 2 \bullet A_i \bullet 3 - 1$, $d_{i+1}$ будет "+". При $C_{i+1} < 0$ если $C_{i+1} < -A_i \bullet 3$. и $B_{i+1} = C_{i+1} + 2 \bullet A_i \bullet 3 - 1$, $d_{i+1}$ будет "–".

Рассмотрим случай $|C_{i+1}| = A_i \bullet 3$, тогда $B_{i+1}$ можно представить в виде

$$B_{i+1} = C_{i+1} - 2 \bullet A_i \bullet 3 \bullet d + (½)^2 - ¼$$

где d равняется ½ для $C_{i+1} > 0$ и –½ для $C_{i+1} < 0$. Поэтому значение $d_{i+1}$ зависит от значения $n_{i+1}$ необработанной части числа N, рассматриваемой как число с точкой перед первым тритом.

То, что функция квадратного корня возрастающая, позволяет определить значение $d_{i+1}$. При $C_{i+1} > 0$ и $n_{i+1} > ¼$ невычисленная часть корня больше ½, т.е. $d_{i+1}$ будет “+”. При $C_{i+1} < 0$ и $n_{i+1} <¼$ невычисленная часть корня меньше –½, т.е. $d_{i+1}$ будет “-”.

### Операция потритного сравнения и сложение трех слагаемых

Введем операцию *потритного сравнения* двух значений. Эта операция проще чем вычитание с последующим определением знака результата. Вычитание выполняется, начиная с младших тритов с возможным переносом единиц (положительных или отрицательных) в старшие триты. Потритное сравнение производится слева направо до первых несовпадающих тритов, сравнение которых и определяет результат. В случае совпадающих значений результат сравнения равен нулю. Приведем обозначения, которые будем использовать при выполнении сравнения

+−0−**+**+−−−
+−0−**−**−+++ >

Выделены первые несовпадающие триты “+” и “–” определяют результат >.

Кроме того будем использовать одновременное *сложение трех слагаемых*, заметим, что в этом случае перенос в старший трит, также как и при обычном сложении, может быть только одной единицы (положительной или отрицательной). Максимальное значение суммы тритов будет при совпадении знаков суммируемых тритов и трита переноса и равно 4 (“++” или “−−”), и это дает перенос единицы того же знака (“+” или “−”) в следующий трит.

Заметим, что трехвходовые сумматоры использовались в «Сетуни» [3] в множительном устройстве. Там для умножения 18–ти тритных чисел на первом уровне использовалось 6 трехвходовые сумматоров, на втором – 2 трехвходовые сумматора для суммирования результатов первого уровня, а третьем уровне - один обычный (двухвходовой) сумматор для суммирования результатов второго уровня. Это позволило при последовательном выполнения операций выполнять умножения за 320 машинных тактов, притом что сложение выполнялось за 180 тактов.

Одновременное сложение трех слагаемых можно использовать при вычислении $B_{i+1}$ . Для случая, когда $d_{i+1}$ равно “+” формулу можно записать в виде

$$B_{i+1} = C_{i+1} +( - A_i \bullet 3) +( - A_i \bullet 3 - 1)$$

Для случая, когда $d_{i+1}$ равно "–", формулу можно записать в виде

$$B_{i+1} = C_{i+1} + A_i \bullet 3 + (A_i \bullet 3 - 1)$$

Так в первом рассмотренном ниже примере на третьем шаге имеем

| | | |
|---|---|---|
| $C_{i+1}$ | −++0 | < |
| $A_i$ | +0 | |

Так как $C_{i+1} < 0$, то сравнение производится с $-A_i \bullet 3$

| | | |
|---|---|---|
| −++0 | | |
| 0−00 | < | т.е. $d_{i+1}$ есть "−". |

Вычисление $B_{i+1}$

| | |
|---|---|
| $C_{i+1}$ | −++0 |
| $A_i \bullet 3$ | +00 |
| $A_i \bullet 3 - 1$ | +0− |
| | 00+− |

Заметим, что в случае $|C_{i+1}| = A_i \bullet 3$ и $d_{i+1}$ отличном от нуля, значение $B_{i+1}$ есть

$-A_i \bullet 3 - 1$ при $d_{i+1}$ равном "+"

и $A_i \bullet 3 - 1$ при $d_{i+1}$ равном "−",

т.е. в этих случаях не требуется производить сложение.

**Примеры вычисления квадратного корня**

Рассмотрим два примера вычисления квадратного корня, в которых встречаются все возможные варианты.

В примере 1 подкоренное число +–++0–+0+ содержит нечетное число тритов и после разбивки на пары (пробел разделитель пары) + –+ +0 –+ 0+ первая пара состоит из одного трита, поэтому первый трит корня будет "+" , и на первом шаге имеем

| | |
|---|---|
| + | Корень + |
| – | |
| 0 | |
| 0↓–+ | |

Комбинация ↓–+ указывает приписывание к сумме следующей пары тритов **–+.** В дальнейшем для сокращения записи будем отображать следующим образом

0↓–+

На втором шаге подкоренное число –+ отрицательно (<), поэтому его надо сравнивать со сдвинутым влево значением корня со знаком минус

–+
–0

Результат сравнения будет >, т.е. очередной трит корня будет “0”.

```
−+                    0↓–+        <      Корень   + 0
–0    >
-----------------------------------------------------------
                        –+↓+0
```

На третьем шаге подкоренное число –++0 отрицательно (<), поэтому его надо сравнивать со сдвинутым влево значением корня со знаком минус

–++0
0–00

Результат сравнения будет <, т.е. очередной трит корня будет “-”.

```
−++0                          −++0        <      Корень  +0–
0−00       <                   +00
                               +0−
-----------------------------------------------------------
                              00+−↓−+
```

На четвертом шаге очередной трит корня будет “0”, т.к. подкоренное число положительно (>), а сравнение его со сдвинутым влево значением корня дает <. На пятом шаге подкоренное число положительно (>), а сравнение его со сдвинутым влево значением корня дает > поэтому очередной трит корня будет “+”.

**Пример 1**

```
    Подкоренное число  +−++0−+0+   = 5329
        Разбиение на пары   +  −+  +0−  +  0+
                            −                    Корень  +
      ------------------------------------------------
                            0↓−+
−+                          0−+           <      Корень  +0
−0              >
        ------------------------------------------------
                            −+↓+0

                            −++0
−++0                        −++0          <      Корень  +0−
0−00     <                   +00
                             +0−
          ------------------------------------------------
                            00+−↓−+
```

```
+−−+                        +−−+          >        Корень  +0−0
+0−0    <

                        +−−+↓0+
                        +−−+0+
+−−+0+                  +−−+0+        >      Корень  +0−0+
0+0−00   >               −0+00
                         −0+0−
   ____________________________________________________
                        000000

                                           Корень  +0−0+ = 73
```

В примере 2 подкоренное число +0+00+ и после разбитие на пары тритов первая слева пара состоит из двух тритов

+0 +0 0+

Поэтому надо добавить еще одну нулевую пару слева

0 +0 +0 0+

Сравнение подкоренного числа с ¼ дает >, т.е. первый трит корня будет “+”, и на первом шаге имеем

```
                                          0
   +0+00+                                 −                 Корень  +
¼   +−+−    .  .  . >
___________________________________________________________
                                          −↓+0
```

На втором шаге подкоренное число –+0 отрицательно (<), поэтому его надо сравнивать со сдвинутым влево значением корня со знаком минус

```
−+0
0−0
```

Результат сравнения будет <, т.е. очередной трит корня будет “-”, и имеем

```
   −+0                      −↓+0        <          Корень  +–
   0−0       <                +0
                              +−
___________________________________________________________
                          00−↓+0
```

На третьем шаге подкоренное число –+0 отрицательно (<), поэтому его надо сравнивать со сдвинутым влево значением корня со знаком минус

```
+0
−+0
```

Результат сравнения будет = и для определения трита зависит надо сравнить необработанную часть подкоренного числа 0 + с ¼. Результат сравнения

```
+0
+– . . . <
```

и трит корня будет “-”.

Если бы на этом шаге необработанная часть вместо “0+” была бы “++”, то сравнение с ¼ дало бы

```
++
+–+– . . . >
```

и трит корня был бы “0”.

На последнем рассмотренном шаге трит корня будет “+”.

**Пример 2**

```
   Подкоренное число            +0+00+    = 271
   Разбиение на пары            +0 +0 0+
   Добавление нулевой пары   0  +0 +0 0+
  +0+00+                    −                  Корень +
¼ +−+−   . . . >
_______________________________________________
                            −↓+0
−+0                         −+0        <       Корень +−
0−0             <            +0
                            +−
_______________________________________________
                           00−↓+0
 −+0                       00−+0       <       Корень +−−
 −+0     =                    +−0

  0+                          +−−
¼ +−+−. . .     <
_______________________________________________
                            +−−↓0+
+−−0+                       +−−0+       >      Корень +−−+
0+−−0           >            +−−0
                             +−−−
                            +0−+0
_______________________________________________
                                     Корень +−−+= 16
```

## Литература


1. Shelburne Br.J. Balanced (Signed) Ternary Notation http://userpages.wittenberg.edu/bshelburne/BalancedTernaryTalkSu09.pdf
2. Рамиль Альварес Х. Алгоритмы деления и извлечения квадратного корня в троичной симметричной системе // Вестн. Моск. ун –

та. Сер. 15. Вычислительная математика и кибернетика. 2008. № 2. С. 42 –45. Перевод на англ. яз. Algorithms for division and extraction of square root in the balanced ternary system // Moscow University Computational Mathematics and Cybernetics. Allerton Press, 2008. V. 32, n.2. P. 103–107.

3. Брусенцов Н.П., Маслов С.П. и др. Малая цифровая вычислительная машина "Сетунь". – М.: Изд–во МГУ, 1965.

Опубликовано: Программные системы и инструменты. Тематический сборник № 11. М.: Изд-во факультета ВМиК МГУ, 2010. С. 98-106.