

# Десятичная арифметика в двоичных вычислениях с фиксированной точкой.

engi@sumus.team

24 июня 2018 г.

## Аннотация

В некоторых областях применения вычислительной техники традиционно используется десятичная система счисления. Сама вычислительная техника, в подавляющем большинстве аппаратуры и моделей программирования, использует системы счисления, базирующиеся на двоичной системе счисления. Рассмотрим некоторые проблемы и их решения, связанные с переходом представления чисел в десятичной системе счисления к вычислениям в двоичной системе счисления. Будут рассмотрены 4 арифметических действия: сложение, вычитание, умножение и деление с округлением.

## 1 Запись и внутреннее представление числа.

Запись числа, как десятичного, является ещё и набором методов алгоритмов и правил для выполнения арифметических действий. Например, правило сложения многоразрядных чисел, “деление в столбик” и другие алгоритмы для десятичных чисел. Вычислительные машины используют двоичную систему счисления и отличающиеся от десятичных правила и алгоритмы.

## 2 Вычисления с целыми числами.

Запись целого неотрицательного числа в любой позиционной системе счисления:

$$x = \sum_{k=0}^{n-1} a_k b^k \quad (1)$$

При выполнении операций  $(+, -, *)$  не возникает проблем перевода чисел в любую другую систему счисления. Деление с округлением подразумевает получение дробной части, возможно, бесконечной периодической. Такая часть для систем счисления с чётным основанием округляется так: если первая цифра дробной части меньше половины основания системы счисления (1 для двоичной, 5 для десятичной) то целая часть результата остаётся без изменений, в другом случае - целая часть результата увеличивается на 1.

Запись отрицательных чисел выглядит как префиксирование числа знаком “ $-$ ”. Внутренним представлением целого отрицательного числа в большинстве моделей программирования будет “дополнительное”. При таком представлении, сложение и вычитание чисел со смешанными знаками выглядят как те же действия с неотрицательными числами. Переполнение результата выглядит как неожидаемый знак результата. Умножение и деление чисел со смешанными знаками в “дополнительном” представлении требует приведения их неотрицательный вид, а затем восстановление знака в результате.

## 3 Вычисления с числами с фиксированной точкой.

Есть несколько подходов для работы с числами десятичной записи:

- Двоично-десятичное представление.
- Двоичное представление.
- Двоичное представление с контролем дробной части.

### **3.1 Двоично-десятичное представление.**

Такое представление максимально приближено к целевому. Соответственно, при таком представлении не возникает проблем перевода между системами счисления для записи. Усложнением при работе с таким представлением будет отсутствие двоично-десятичной арифметики в современных моделях программирования. Выполнение действий нужно будет производить индивидуально по каждой паре цифр, что существенно увеличивает вычислительные затраты. Кстати, большинство “бухгалтерских” калькуляторов работает именно в таком представлении чисел.

### **3.2 Двоичное представление.**

Это представление максимально приближено к “машинному”. Все внутренние операции выполняются минимальным количеством машинных действий. При попытке совместить двоичное представление и десятичную запись чисел с дробной частью возникают проблемы. Например, 0.1 десятичной записи, в двоичной записи с фиксированной точкой превратится в бесконечную периодическую дробь 0.0(0011), что безусловно приводит к потере точности. Подобную ситуацию можно “залатать” алгоритмом с избыточной точностью представления/вычисления и предварительным округлением при записи числа в десятичном<sup>1</sup>. Однако, такой метод не решает проблему полностью. При определённых вычислениях, предварительно округлённые в одну сторону числа могут дать в результате число, отличающееся на 1 в младшем разряде.

### **3.3 Двоичное представление с контролем дробной части.**

Предлагается представление числа, аналогично двоичному, но с набором коррекций по всем арифметическим действиям. Дополнительным условием упрощения набора операций будет использование дробных чисел единого формата, с десятичной точкой перед фиксированным количеством цифр. Для числа с фиксированной десятичной точкой при известном количестве цифр в дробной части введём скрытый делитель:

$$x = X/D$$
$$D = 10^d$$

где:

$x$  — число с фиксированной десятичной точкой.

$X$  — целое число.

$D$  — делитель.

$d$  — количество цифр после десятичной точки.

#### **3.3.1 Приведение десятичной записи.**

Для перевода числа из десятичной записи:

1. Добавить нули справа до получения количества цифр после десятичной точки.
2. Перевести число из десятичной записи как целое стандартным преобразованием.

Для перевода числа в десятичную запись:

1. Перевести число в десятичную запись как целое стандартным преобразованием.
2. “Поставить” десятичную точку в позицию справа и отбросить нули справа.

#### **3.3.2 Приведение целого числа к дробному виду.**

Чтобы привести целое число к дробному виду нужно умножить его на делитель .

#### **3.3.3 Округление дробного числа.**

Для округления дробного числа до указанного количества цифр после десятичной точки нужно:

1. Разделить на с округлением результата.
2. Умножить на .

---

<sup>1</sup>Есть подозрение, что в Excel так и сделано.

### **3.3.4 Сложение/вычитание.**

Сложение/вычитания двух дробных чисел производится также как и целых чисел. Сложение/вычитания дробного и целого чисел требует приведения целого числа к дробному виду.

### **3.3.5 Умножение.**

Формула:

Показывает необходимость приведения результата. Умножение выполняется так:

1. Перемножить представления чисел с получением результата удвоенной разрядности.
2. Произвести целочисленное деление с округлением предыдущего результата на .
3. Отбросить “лишние” старшие разряды для получения требуемой разрядности представления дробного числа.

### **3.3.6 Деление.**

Формула:

Показывает ход выполнения действий. Деление выполняется так:

1. Помножить делимое на с получением результата увеличенной разрядности.
2. Произвести целочисленное деление с округлением – предыдущего результата на делитель.
3. Отбросить “лишние” старшие разряды для получения требуемой разрядности представления дробного числа.

## **4 Реализация.**

Современная реализация модели “двоичное представление с контролем дробной части” сделана в Java-классе `java.math.BigDecimal`. Удобными возможностями этого класса являются индивидуальное задание десятичной точки (scale в терминологии библиотеки) и широкий диапазон представляемых чисел. Представляется несложным записать набор классов/функций выполняющих требуемые операции для чисел с предопределенным форматом.

## **5 Выполнение операций с длинными целыми.**

### **5.1 Представление.**

Представление длинного целого – это массив целых примитивов, например, массив `long`.

### **5.2 Знак числа и смена знака.**

Признаком отрицательного числа будет “1” в старшем бите старшего слова. Ноль не отличается по знаку от положительных чисел. Для смены знака:

- Бит-инверсия.
- Инкрементирование.

**ВНИМАНИЕ!** При смене знака максимального по модулю отрицательного числа происходит переполнение.

## **5.3 Сложение и вычитание.**

### **5.3.1 Сложение.**

Сложение по словам от младшего к старшему, с учётом переноса. Перенос при сложении двух целых одной разрядности и бита переноса из предыдущей суммы можно вычислить так:

- Если оба старших бита “1”, то перенос будет.
- Если оба старших бита “0”, то переноса не будет.
- Если только один старший бит “1”, то перенос будет если старший бит суммы “0” и не будет переноса, если старший бит суммы “1”.
- Перенос “снизу” при сложении младших слов равен “0”.
- Перенос “вверх” от сложения старших слов игнорируется.

### **5.3.2 Вычитание.**

Для вычитания можно использовать два подхода:

- Использовать сложение со сменой знака вычитаемого.
- Произвести вычитание с введением дополнительной базовой функции.

Предлагаю использовать минимум базовых функций, соответственно “прямое” вычисление разности не приводится.

### **5.3.3 Переполнение результата при сложении и вычитании.**

Переполнение результата при сложении двух целых одного знака – это смена знака результата. Переполнение результата при сложении двух целых разных знаков не возникает. Так как вычитание производится с помощью сложения, то переполнение результата происходит, так же как и при сложении.