Сложение и вычитание целых чисел с фиксированной запятой. Сложение и вычитание чисел с плавающей запятой

Любая информация (числа, команды, записи и т. п.) представляется в ЭВМ в виде двоичных кодов фиксированной или переменной длины. Отдельные элементы двоичного кода, имеющие значение 0 или 1, называют разрядами или битами. Двоичный код, состоящий из 8 разрядов, носит название байта. Для записи чисел также используют 32-разрядный формат (машинное слово), 16-разрядный формат (полуслово) и 64-разрядный формат (двойное слово).

В ЭВМ с целью упрощения выполнения арифметических операций применяют специальные коды для представления чисел. Использование кодов позволяет свести операцию вычитания чисел к арифметическому сложению кодов этих чисел, при этом упрощается определение знака результата операции и облегчается выработка признаков переполнения разрядной сетки. Применяются прямой, обратный и дополнительный коды чисел. В результате упрощаются устройства ЭВМ, выполняющие арифметические операции.

К кодам предъявляются следующие требования.

1. Разряды числа в коде жестко связаны с определенной разрядной сеткой.
2. Для записи знака кода в разрядной сетке отводится фиксированный, строго определенный разряд, т.е. определяется общая длина кода, в котором выделяются цифровые разряды и знаковый (крайний слева) разряд, представляющий знак числа, причем знак «+» кодируется цифрой 0, а знак «-» - цифрой 1.

Для представления **отрицательных** чисел в ЭВМ применяют прямой, обратный и дополнительный коды. **Положительные числа** представляют в прямом коде.

Выполнение арифметических операций:

- сложение;
- вычитание;
- умножение.

для двоичных чисел в форме :

- с фиксированной запятой;
- с плавающей запятой.

Представление чисел в формате с фиксированной запятой

Целые числа в компьютере хранятся в памяти в формате *с фиксированной запятой*. В этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа, а запятая находится справа после младшего разряда, т.е. вне разрядной сетки.

Для хранения *целых неотрицательных чисел* отводится одна ячейка памяти (8 бит). Например, число $A_2 = 10101010_2$ будет хранится в ячейке памяти следующим образом:

1	0	1	0	1	0	1	0

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех ячейках хранятся единицы. Для n-разрядного представления оно будет равно:

$$2^n - 1$$

Диапазон изменения *целых неотрицательных чисел* от 0 до 255.

Для хранения *целых чисел со знаком* отводится две ячейки памяти (16 бит), причем старший (левый) разряд отводится под знак числа (если число положительное, то в знаковый разряд записывается 0, если число отрицательное записывается 1).
Представление в компьютере положительных чисел с использованием формата «знак-величина» называется *прямым кодом* числа. Например, число $2002_{10} = 11111010010_2$ будет представлено в 16-ти разрядном представлении следующим образом:

0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0

При представлении целых чисел в n-разрядном представлении со знаком максимальное положительное число (с учетом выделения одного разряда на знак) равно:

$$A = 2^{n-1} - 1$$

$$A_{10} = 2^{15} - 1 = 3276710$$

Для получения дополнительного кода отрицательного числа можно использовать довольно простой алгоритм:

1. Модуль числа записать *прямым кодом* в *n* двоичных разрядах;
2. Получить *обратный код* числа, для этого значения всех бит инвертировать (все единицы заменить на нули и все нули заменить на единицы);
3. К полученному *обратному коду* прибавить единицу.

Пример. Записать дополнительный код отрицательного числа –2002 для 16-ти разрядного компьютерного представления с использованием алгоритма.

Прямой код	$\lvert -2002_{10} \rvert$	0000011111010010_2
Обратный код	инвертирование	1111100000101101_2
	прибавление единицы	1111100000101101_2 $+0000000000000001_2$
Дополнительный код		1111100000101110_2

Сложение двоичных чисел с фиксированной запятой

Операция сложения двух чисел (целых иди дробных) с фиксированной запятой с произвольными знаками может выполняться в ЭВМ:

- в прямых кодах (для положительных чисел);
- в обратном или дополнительном кодах (для отрицательных чисел).

При алгебраическом сложении чисел с фиксированной запятой положительные числа остаются в прямом коде, а отрицательные числа преобразуются в обратный или дополнительный код.

При сложении чисел в ЭВМ используют правила сложения двоичной арифметики: 0+0=0; 1+0=1; 0+1=1; 1+1=10.

Знаковый разряд участвует в суммировании как и значащие. При возникновении единицы переноса из знакового разряда для дополнительных кодов, она отбрасывается, а для обратных кодов прибавляется к младшему разряду суммы (циклический перенос).

Положительная сумма получается в прямом коде, а отрицательная – в коде представления слагаемых.

Переполнение разрядной сетки

В результате выполнения операции сложения может получиться результат, превышающий максимально возможное число для заданной разрядной сетки, т.е. происходит выход полученного результата за пределы разрядной сетки в сторону знакового разряда, называемый «**переполнением**» (значащий разряд становится знаковым, результат операции неверный).

Пример 1. Найти сумму $(A_1+A_2)_{доп} = A_{1\ доп} + A_{2\ доп.}$

$A_{1\ доп}$ = 1 01001 (целое десятичное число «-23»).

$A_{2\ доп}$ = 1 01110 (целое десятичное число «-18»).

Суммируем числа в дополнительном коде:

```
  1 01001
+ 1 01110
  0 10111
```

Единица переноса из знакового разряда игнорируется. Результат машиной ошибочно воспринимается как положительное число.

Способы выявления переполнения в арифметических операциях:

1) Анализируются два переноса – из старшего значащего разряда в знаковый (p1) и из знакового разряда (p2).

Если есть оба переноса или нет ни одного переноса, то переполнения нет, если есть только один из переносов, то имеет место переполнение.

Сигнал ф («Останов») будет вырабатываться по формуле:

$$\varphi = \begin{cases} 1, \textit{если} \quad p1 \neq p2, \\ 0, \textit{если} \quad p1 = p2. \end{cases}$$

Способы выявления переполнения в арифметических операциях:

2) Для представления чисел применяют модифицированный код. На переполнение при сложении двух чисел указывают несовпадение цифр в знаковых разрядах результата. Комбинация 01 соответствует переполнению положительного результата, а комбинация 10 – отрицательного.

Пример 2. Найти сумму двух чисел с фиксированной запятой, представленных в дополнительном коде:

$A_{1\ доп}$ = 1 01001 (целое десятичное число «–23»);

$A_{2\ доп}$ = 1 01110 (целое десятичное число «–18»).

Для приведенного примера суммируем числа в доп. модиф. коде:

$$\begin{array}{r} 11\ 01001 \\ +\ 11\ 01110 \\ \hline 10\ 10111. \end{array}$$

Комбинация «10» в знаковых разрядах результата является признаком отрицательного переполнения результата.

Способы выявления переполнения в арифметических операциях:

Пример 3. Найти сумму двух чисел с фиксированной запятой, представленных в дополнительном коде:

$A_{1\ доп}$ = 0 01001 (+23);

$A_{2\ доп}$ = 0 01110 (+18).

Для приведенного примера суммируем числа в пр. модиф. коде:

$$\begin{array}{r} 00\ 10111 \\ +\ 00\ 10010 \\ \hline 01\ 01001. \end{array}$$

Комбинация «01» в знаковых разрядах указывает на переполнение положительного результата суммирования.

Способы выявления переполнения в арифметических операциях:

Результат обеих операций неверный и дальнейшее решение задачи не имеет смысла.

ЭВМ вырабатывает сигнал $\varphi = 1$ (Останов).

Если сигнал $\varphi = 0$, то переполнения нет, результат верный и можно продолжить решение задачи.

Если обозначить знаковые разряды: младший разряд – зн 1, старший разряд – зн 2, то значение сигнала φ будет вырабатываться по формуле:

$$\varphi = \begin{cases} 1, \textit{если } зн1 \neq зн2, \\ 0, \textit{если } зн1 = зн2. \end{cases}$$

Алгоритм сложения чисел с фиксированной запятой

Алгоритм сложения:

1. Положительные числа остаются без изменения (в прямом коде), отрицательные числа переводятся в дополнительный код.

2.Суммируются полученные коды чисел, включая знаковые разряды. Если имеет место перенос из знакового разряда, он отбрасывается.

3.Анализируется результат (сумма) на переполнение. Если имеет место переполнение, то вырабатывается сигнал $\varphi = 1$ и ЭВМ останавливает решение задачи.

4.Если переполнения нет, то анализируется результат по знаковому разряду: 0 – результат в прямом коде, 1 – результат в дополнительном коде.

Пример сложения чисел с фиксированной запятой

Пример 4. Заданы числа. Выполнить операцию сложения ($A_1 + A_2$).

$[A_1]_{пр} = 1\ 0110$, $[A_1]_{доп} = 1\ 1010$.

$[A_2]_{пр} = 1\ 1001$, $[A_2]_{доп} = 1\ 0111$.

Суммируем $[A_1]_{доп} + [A_2]_{доп}$:

$$\begin{array}{rl} & 1\ 1010_{доп} \\ + & \underline{1\ 0111_{доп}} \\ & 1\ 0001_{доп}. \end{array}$$

Единица переноса из знакового разряда в результате игнорируется. Результат отрицательный и представлен в дополнительном коде. Переполнения нет, так как присутствуют оба анализируемых переноса.

Результат: $[A_{рез}]_{доп} = 1\ 0001$; $[A_{рез}]_{пр} = 1\ 1111$.

Проверка: (–6) + (–9) = (–15).

Вычитание двоичных чисел с фиксированной запятой

Операция вычитания чисел (целых или дробных**)** заменяется суммой:

$$[A_1]_{пр} - [A_2]_{пр} = [A_1]_{пр} + [-A_2]_{пр}.$$

Знак вычитаемого в прямом коде инвертируется. После этого выполняется операция сложения уменьшаемого и вычитаемого по алгоритму с использованием дополнительного кода для представления отрицательных слагаемых.

Пример. Выполнить арифметическое действие 3000_{10} - 5000_{10} в 16-ти разрядном компьютерном представлении.

Представим положительное число в прямом, а отрицательное число в дополнительном коде:

Десятичное число	Прямой код	Обратный код	Дополнительный код
3000	0000101110111000		
-5000	0001001110001000	1110110001110111	1110110001110111 +0000000000000001 1110110001111000

Сложим прямой код положительного числа с дополнительным кодом отрицательного числа. Получим результат в дополнительном коде:

3000-5000			1111100000110000

Переведем полученный дополнительный код в десятичное число:

1) Инвертируем дополнительный код: 0000011111001111
2) Прибавим к полученному коду 1 и получим модуль отрицательного числа:

```
 0000011111001111
+0000000000000001
-----------------
 0000011111010000
```

3) Переведем в десятичное число и припишем знак отрицательного числа: -2000.

Представление чисел в формате с плавающей запятой

Вещественные числа (конечные и бесконечные десятичные дроби) хранятся и обрабатываются в компьютере в формате *с плавающей запятой*. В этом случае положение запятой в записи числа может изменяться.

Формат чисел *с плавающей запятой* базируется на экспоненциальной форме записи, в которой может быть представлено любой число.

Представление двоичных чисел с плавающей запятой

Число А в форме с плавающей запятой представляется в виде

$$A = m_n \cdot q^P ,$$

где m_n – нормализованная мантисса числа А;

P – порядок (характеристика) числа А;

q – основание системы счисления.

Мантисса m_n представляет собой правильную дробь, удовлетворяющую условию

$$q^{-1} \le | m_n | < 1.$$

Числа A_1 и A_2 представлены следующим образом:

$$A_1 = m_1 \cdot q^{P1}; \quad A_2 = m_2 \cdot q^{P2}.$$

Арифметическое сложение или вычитание мантисс двух чисел может быть выполнено только в случае равенства их порядков.

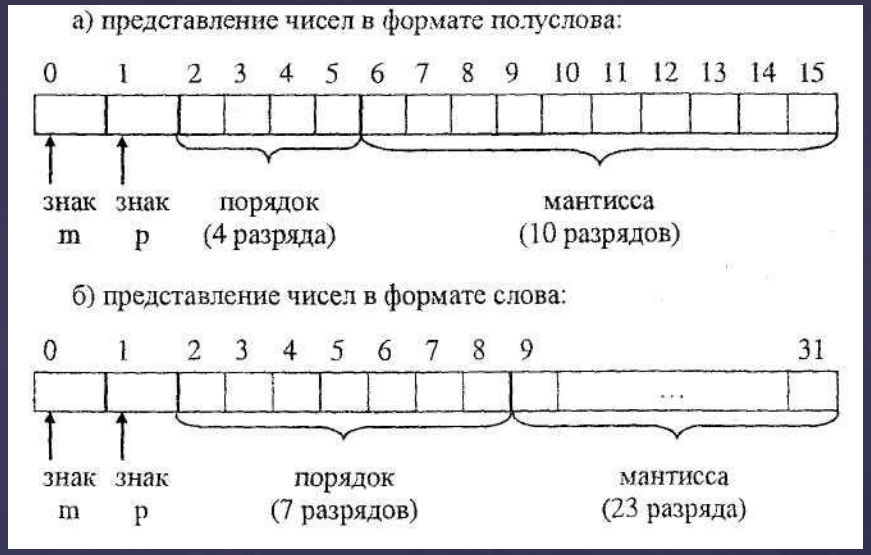
Таким образом, в нормализованных числах первая цифра после точки должна быть значащей:

Пример. Преобразуйте десятичное число 888,888, записанное в естественной форме, в экспоненциальную форму с нормализованной мантиссой.

$888{,}888 = 0{,}888888 \times 10^3$

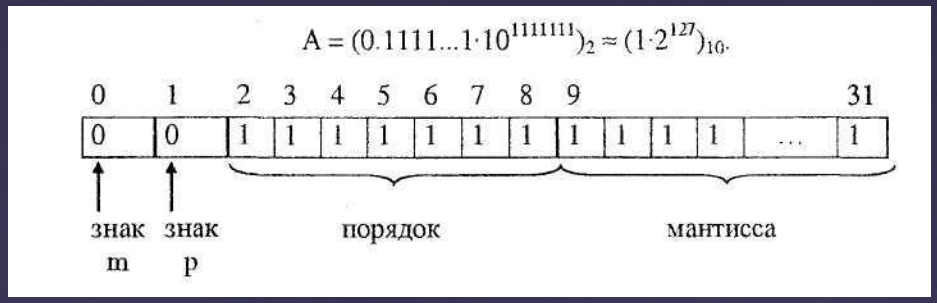
Нормализованная мантисса $m = 0{,}888888$, порядок $n = 3$.

Для представления чисел в машинном слове выделяют группы разрядов для изображения мантиссы, порядка, знака числа и знака порядка:

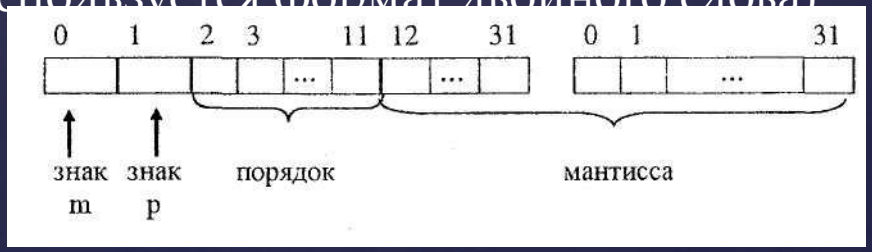


с плавающей точкой в формате 32-разрядного слова будет иметь вид

Максимальным числом, представимым в формате слова, будет число



Числа с плавающей точкой позволяют увеличить диапазон обрабатываемых чисел, но при этом точность их изображения определяется только разрядами мантиссы и уменьшается по сравнению с числами с фиксированной точкой. При записи чисел в формате слова диапазон представимых чисел будет от - $1\text{-}2^{127}$ до 1-2 (2 =10), а точность их представления будет определяться мантиссой, состоящей из 23 разрядов. Точность может быть повышена путем увеличения количества разрядов мантиссы. Это реализуется путем представления чисел с так называемой двойной точностью (используется формат двойного слова):



Алгоритм сложения двух чисел с плавающей запятой:

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Нормализация мантиссы

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Нарушение нормализации мантиссы вправо

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Нарушение нормализации мантиссы влево

Признак нарушения нормализации влево для дополнительных и обратных кодов – это сочетание 01 или 10 в знаковых разрядах модифицированных кодов.

Устранение этого нарушения состоит в модифицированном сдвиге мантиссы M_x вправо на 1 разряд и увеличении порядка P_x на единицу.

Примеры для доп. и обр. кодов:

$M_x = 01,110111, \quad P_x = 00,101$

После нормализации:

$M_x = 00,111011 \quad ①^*, \quad P_x = 00,110$

$M_x = 10,10011, \quad P_x = 00,011$

После нормализации:

$M_x = 11,01001 \quad ①^*, \quad P_x = 00,100$

* – разряд, вышедший за пределы разрядной сетки, используется для округления или отбрасывается.

Пример.

1. Выполнить сложение чисел

$$X=+0.101\cdot10^{+101}$$

$$Y=-0.1011\cdot10^{+100}$$

(количество разрядов порядка Np = 4, мантиссы Nq = 8).

$$p_x - p_y = p_x + (-p_y)$$

а) Выполним выравнивание порядков чисел, для чего вычислим значение разности

p_y = 1100 - прямой код порядка числа Y;

p_y = 1011 - обратный код порядка числа Y;

p_y = 1100 - дополнительный код порядка числа Y.

Находим разность порядков:

$$p_x - p_y = (0101 + 1100)_{доп} = (0001)_{пр}$$

Так как $p_x > p_y$, необходимо сдвинуть мантиссу числа Y на один разряд вправо:

$$Y_{пр} = 1.1011000.$$

Сдвигаем:

$$Y_{пр} = 1.0101100.$$

б) Выполним сложение (вычитание) мантисс:

$$Y_{доп} = 11010011_{обр} + 00000001 = 11010100_{доп}$$

Суммируем мантиссы:

$$q_z = (q_x + q_y) = (01010000_{пр} + 11010100_{доп}) = 00100100_{пр}.$$

в) Порядок результата принимаем равным порядку числа X, т.е.

$$p_z = +101.$$

г) Полученную сумму нормализуем, так как

$$|q_z| < \frac{1}{s}.$$

Для нормализации необходимо p_z уменьшить на 1, а q_z сдвинуть на один разряд влево:

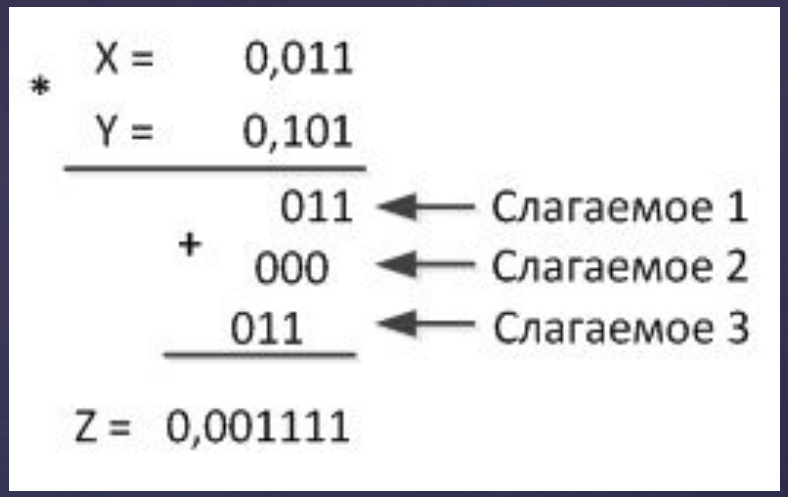
$$Z = 0.1001\cdot10^{+100}.$$

Умножение двоичных чисел в форме с фиксированной запятой

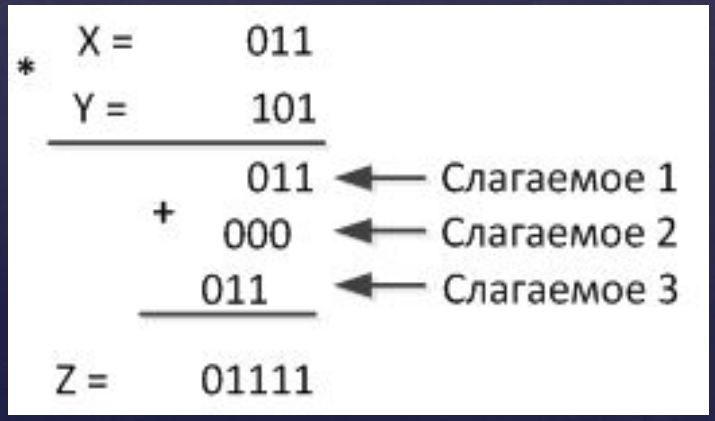
В математике известен метод умножения чисел в столбик. Метод для целых и дробных двоичных чисел.

Умножение, начиная с младшего разряда множителя:

а) дробные числа



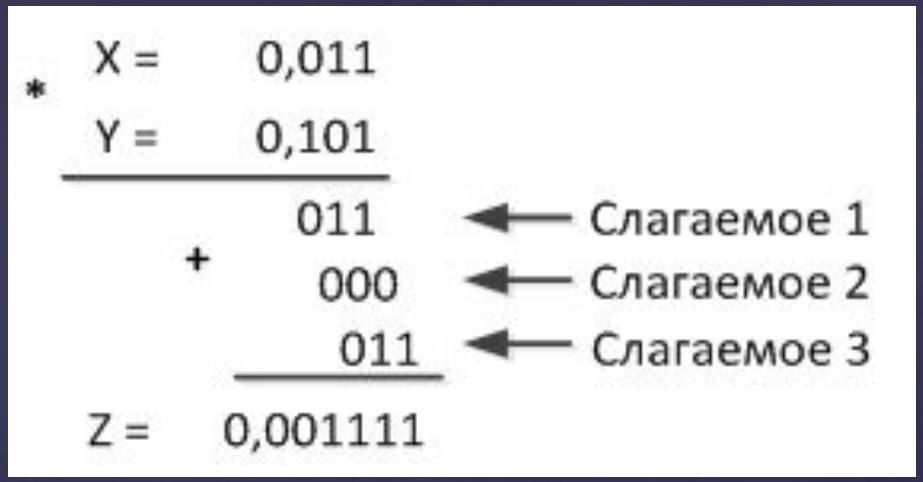
б) целые числа



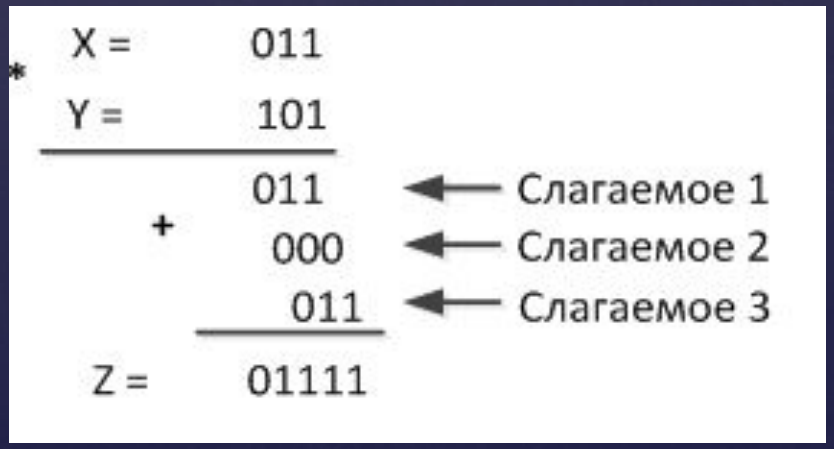
Умножение двоичных чисел в форме с фиксированной запятой

Умножение, начиная со старшего разряда множителя:

а) дробные числа



б) целые числа



Умножение двоичных чисел в форме с фиксированной запятой

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Умножение двоичных чисел в форме с фиксированной запятой

Приведенные правила (1-3) позволяют сформулировать алгоритм пошагового вычисления произведения Z путем отыскания на каждом i-ом шаге частичного произведения $ЧП_i$.

При этом принимается, что на начальном шаге $ЧП_0 = 0$. Число шагов определяется количеством числовых разрядов множителя.

Общая идея алгоритма заключена в вычислении $ЧП_i$ на каждом шаге алгоритма:

Z=	0	+	$Сл_1$		+ $Сл_2$		+ …….……		+ $Сл_n$	
	$ЧП_0$									
1° шаг	$ЧП_0$	+	$Сл_1$	=	$ЧП_1$					
2° шаг					$ЧП_1$	+	$Сл_2$	=	$ЧП_2$	
……							$ЧП_2$	+	$Сл_3$	……
n° шаг	……		……						$ЧП_{n-1}$ + $Сл_n$	= $ЧП_n$

Умножение двоичных чисел в прямом коде

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Алгоритм умножения операндов в прямых кодах

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Пример умножения операндов в прямых кодах

Пример 1. Умножение в прямом коде $Z_{пр} = A_{пр} * B_{пр}$.

$A_{пр} = 10110 = (–6)_{10}$;

$B_{пр} = 11101 = (–13)_{10}$.

В этом случае перемножаются модули чисел, а произведению присваивается знак «плюс», если знаки сомножителей одинаковы, или знак «минус», если знаки разные.

|A| =00110; |B| = 01101.

Перемножаем числа целые, следовательно произведение должно быть представлено двойной (2n) разрядностью.

Пример умножения операндов в прямых кодах

* 00110
  01101
  0000000000 – $\Sigma ЧП_0$;
+ 00110 – прибавление множимого, разряд множителя равен 1;
  0011000000 – $\Sigma ЧП_1$;
  0001100000 – сдвиг вправо на 1 разряд $\Sigma ЧП_1$;
  0000110000 – сдвиг вправо на 1 разряд $\Sigma ЧП_2$;
+ 00110 – прибавление множимого, разряд множителя равен 1;
  0011110000 – $\Sigma ЧП_3$;
  0001111000 – сдвиг вправо на 1 разряд $\Sigma ЧП_3$;
+ 00110 – прибавление множимого, разряд множителя равен 1;
  0100111000 – $\Sigma ЧП_4$;
  0010011100 – сдвиг вправо на 1 разряд $\Sigma ЧП_4$;
  0001001110 – дополнительный сдвиг вправо на 1 разряд после умножения на все значащие разряды множителя для правильной постановки результата в формате 2n разрядов (или умножение на знаковый разряд).

$|A| = 00110$; $|B| = 01101$.

Алгоритм умножения операндов в прямых кодах

Одновременно с умножением на знаковый разряд определяется знак произведения, как «сумма по модулю 2» знаков сомножителей: $Z_{знак} = 0$.

Произведение A* B = $(0001001110)_{2пр} = (+1001110)_2 = (78)_{10}$.

Проверка: $(–6)_{10} * (–13)_{10} = (78)_{10}$.

Алгоритм умножения операндов в дополнительных кодах

1. Выравнивание порядков суммируемых чисел:

а) вычитание из порядка числа A_1 порядка числа A_2 с целью определения, порядок какого числа больше и на сколько:

$P_1 - P_2 = \Delta P$;

б) мантисса числа с меньшим порядком сдвигается вправо на величину разности порядков ΔP.

Если разность порядков положительная, то сдвигается мантисса второго числа, если отрицательная, то сдвигается мантисса первого числа. Обоим числам присваивается больший порядок.

2. Алгебраическое суммирование мантисс (аналогично суммированию дробных чисел с фиксированной запятой).

3. Результату присваивается больший порядок.

4. Проверка на нормализацию результата. При выполнении операции сложения нарушение нормализации суммы может быть влево только на один разряд ($|m_{ns}|<1$) или вправо на любое число разрядов ($q^{-1} \le |m_{ns}|$).

Алгоритм умножения операндов в дополнительных кодах

Следует обратить внимание:

1)слагаемые $Сл_i$ представляются дополнительным кодом без удвоения количества числовых разрядов, так как младшая часть разрядов всегда нулевое двоичное слово аналогично, как и для прямых кодов;

2)при использовании дополнительных кодов частичное произведение $ЧП_i$ получается за счет модифицированного сдвига кода $\Sigma ЧП_{i-1}$ в отличие от простого (арифметического) сдвига для случая использования прямых кодов. Модифицированный сдвиг заключается в размножении знакового разряда;

3)напомним, что корректирующая поправка вводится только при наличии отрицательного множителя.

Пример умножения операндов в дополнительных кодах

Пример 2. Умножение в дополнительном коде $Z_{доп} = A_{доп} * B_{доп}$.

$A_{пр} = (+3)_{10} = 0\ 011_{пр} = 0\ 011_{доп} = 00\ 011^{м}_{доп}$;

$B_{пр} = (–5)_{10} = 1\ 101_{пр} = 1\ 011_{доп} = 11\ 011^{м}_{доп}$.

Так как $B_{пр}<0$, то на последнем шаге требуется ввод корректирующей поправки $K = [-A]^{м}_{доп}$. Для отыскания коэффициента K необходимо выполнить последовательность преобразований:

$[A]^{м}_{пр} \rightarrow [-A]^{м}_{пр} \rightarrow [-A]^{м}_{доп}$. Таким образом, получается:

$00\ 011 \rightarrow 11\ 011 \rightarrow 11\ 101$. Отсюда $K = [-A]^{м}_{доп} = 11\ 101$.

Перемножаем числа целые, следовательно, произведение должно быть представлено двойной (2n) разрядностью.

Пример умножения операндов в дополнительных кодах

* 00 011
  00 011

  00 000000 – $\Sigma ЧП_0$;
+ 00 011 – прибавление множимого, разряд множителя равен 1;
  00 011000 – $\Sigma ЧП_1$;
  00 001100 – сдвиг вправо на 1 разряд $\Sigma ЧП_1$;
+ 00 011 – прибавление множимого, разряд множителя равен 1;
  00 100100 – $\Sigma ЧП_2$;
  00 010010 – сдвиг вправо на 1 разряд $\Sigma ЧП_2$;
  00 001001 – сдвиг вправо на 1 разряд $\Sigma ЧП_3$;
+ 11 101 – корректирующая поправка $K = [-A]^м_{доп}$;
  11 110001 – результат умножения $Z^м_{доп}$.

Произведение $A^м_{доп} * B^м_{доп} = (11\ 110001)_{доп} = (11\ 001111)_{пр} = (-15)_{10}$.

Проверка: $(+3)_{10} * (–5)_{10} = (-15)_{10}$.

Умножение и деление операндов с плавающей запятой

Числа с плавающей запятой A_1 и A_2 представлены следующим образом:

$$A_1 = m_1 \cdot q^{P1}; \quad A_2 = m_2 \cdot q^{P2}.$$

Арифметическое умножение чисел с плавающей запятой сводится к умножению мантисс и сложению порядков (как чисел с фиксированной запятой) :

$$A_1 \cdot A_2 = [m_1 \cdot m_2]; \quad q^{P1+P2}.$$

Арифметическое деление чисел с плавающей запятой сводится к делению мантисс и вычитанию порядков (как чисел с фиксированной запятой) :

$$A_1 : A_2 = [m_1 : m_2]; \quad q^{P1-P2}.$$