ПЕРЕВОД ЧИСЕЛ В ПОЗИЦИОННЫХ СИСТЕМАХ СЧИСЛЕНИЯ

Любое смешанное число A в позиционной системе счисления (СС) с основанием q можно записать:

$$A_q = a_n q^n + a_{n-1} q^{n-1} + \ldots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \ldots + a_{-m} q^{-m} = \sum_{k=-m}^{n} a_k q^k$$

где $a_k < q$ – цифра числа;

q^k – разрядный вес цифры a_k;

$n+1$ – количество разрядов в целой части числа;

m - количество разрядов в дробной части числа.

Для перевода целых чисел и правильных дробей из одной позиционной СС в другую применяются различные правила.

Перевод целых чисел

Пусть p - основание исходной СС, q - основание новой СС, в которую надо перевести целое число A_p. Тогда целое число в СС с новым основанием q можно представить в соответствии с основной формулой:

$$A_p = a_n q^n + a_{n-1} q^{n-1} + \ldots\ldots + a_1 q^1 + a_0.$$

Разделим обе части приведенной формулы на новое основание q:

$$\frac{A_p}{q} = a_n q^{n-1} + a_{n-1} q^{n-2} + \ldots\ldots + a_1 + \frac{a_0}{q}$$

В правой части равенства сформировалась целая часть первого частного и первый остаток от деления a_0 - младшая цифра целого числа в новой СС. Далее целую часть первого частного следует разделить на основание новой СС q, и новый остаток даст вторую искомую цифру a_1 и т.д. Это позволяет сформулировать правило.

Чтобы перевести целое число в новую СС, его надо последовательно делить на основание новой СС до тех пор , пока не получится частное, у которого целая часть равна «0». Число в новой СС записывают из остатков от последовательного деления, причем последний остаток будет старшей цифрой целого числа в новой СС.

Перевод правильных дробей

Пусть p - основание исходной СС, q – основание новой СС. Запишем правильную дробь в СС с новым основанием:

$$A_p = a_1 q^{-1} + a_2 q^{-2} + ... + a_m q^{-m}$$

Умножим обе части равенства на новое основание q:

$$A_p \times q = a_1 + a_2 q^{-1} + ... + a_m q^{-m+1}$$

В правой части равенства a_1 - целая часть первого произведения, являющаяся старшей цифрой дроби в новой СС. Далее, умножив на новое основание q ***дробную часть*** первого произведения, определим вторую цифру дроби a_2 как целую часть второго произведения и т.д. Отсюда следует правило.

Чтобы перевести правильную дробь из одной позиционной СС в другую, её надо последовательно умножать на основание новой СС до тех пор, пока в новой дроби не будет получено требуемого количества цифр, определяемого заданной точностью. Правильная дробь в новой СС записывается из целых частей произведений, и целая часть первого произведения будет старшей цифрой новой дроби.

Перевод дробей - бесконечный процесс и может быть выполнен лишь приближенно. Чтобы сохранить точность исходной дроби, надо определить количество цифр в изображении дроби по новому основанию.

Если m_1 – количество цифр в исходной дроби с основанием p, m_2 – количество цифр в дроби с новым основанием q, то из условия сохранения точности $p^{-m_1} = q^{-m_2}$ можно получить формулу:

$$m_2 = \frac{m_1}{\log_p q} \cong \left]\frac{m_1}{\log_p q}\right[+ 1$$

Далее выполняется округление по последнему разряду, после чего этот последний разряд отбрасывается.

При переводе неправильных дробей отдельно преобразуется целая и дробная части по сформулированным выше правилам, после чего смешанное число записывается в новой системе счисления.

Использование вспомогательных систем счисления

Использование вспомогательных систем счисления позволяет ускорить процесс перевода чисел. При работе с ЭВМ вспомогательные СС имеют основания, кратные степени двойки. Чаще всего используют восьмеричную (8СС) и шестнадцатеричную (16СС) системы счисления.

Для представления любой восьмеричной цифры необходимо три двоичных разряда (триада), для шестнадцатеричной цифры – четыре двоичных разряда (тетрада).

Сформулируем правила перевода чисел из 10СС в 2СС и обратно с использование в качестве вспомогательных 8СС (16СС).

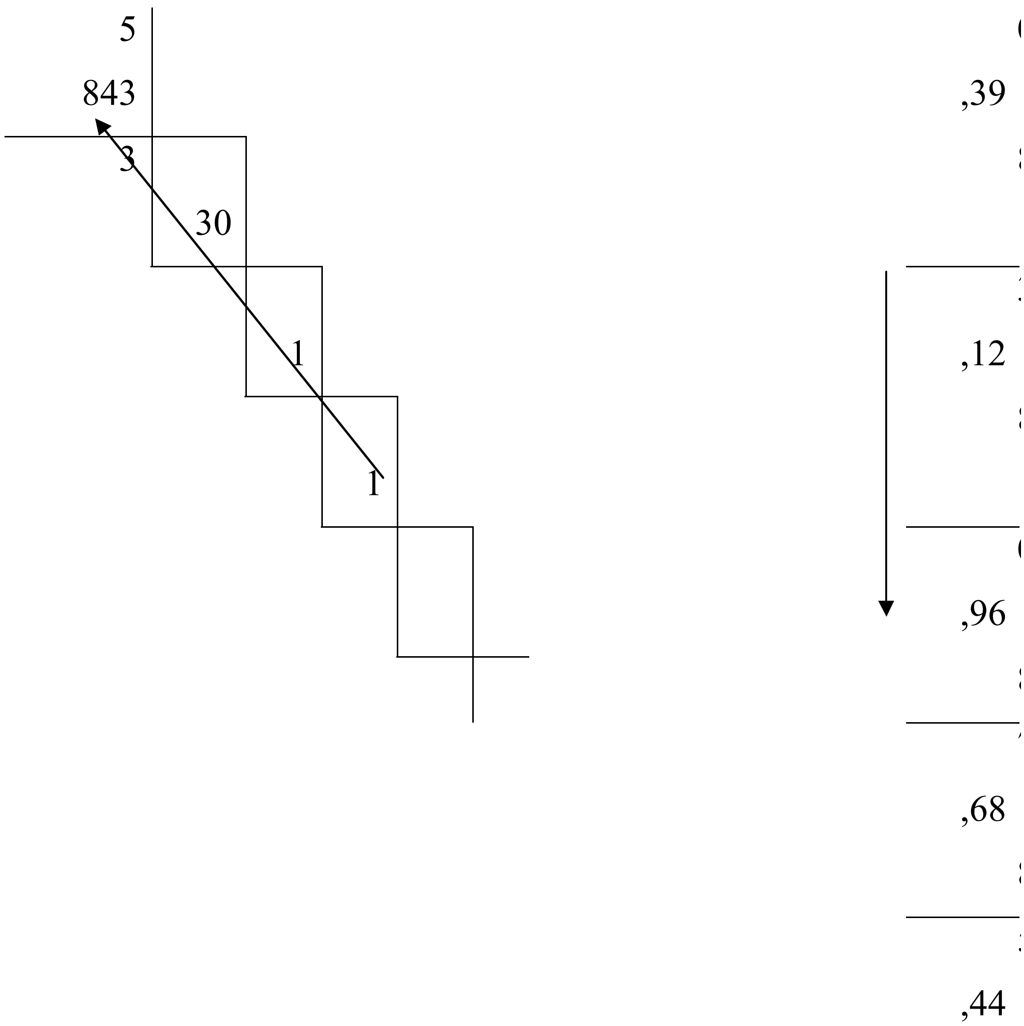
Чтобы перевести число из 10СС в 2СС с использованием 8СС (16СС), надо перевести десятичное число в 8СС (16СС) указанными выше способами, а затем представить цифры восьмеричного (шестнадцатеричного) числа триадами (тетрадами).

Обратный перевод чисел из 2СС в 10СС с использование вспомогательных СС выполняется по следующему правилу.

Вправо и влево от запятой двоичное число разбивается на триады (тетрады), которые заменяются соответствующими восьмеричными (шестнадца-

теричными) цифрами. Далее по основной формуле переходят к 10СС. Причем, если в крайних триадах (тетрадах) недостаточно разрядов, то они дополняются «0»: старшие разряды – слева, младшие – справа.

Пример 1. Число A=5843,39 перевести из 10СС в 2 СС, используя вспомогательную 8СС. Проверить правильность перевода путем преобразования полученного числа из 2СС в 10СС с использованием 16СС.



A=5843,39$_{(10)}$=13323,3075$_{(8)}$=001 011 011 010 011,011 000 111 101$_{(2)}$

Проверка: 0001 0110 1101 0011,0110 0011 1101$_{(2)}$=16D3,63D$_{(16)}$=

$=16^3+6*16^2+13*16+3+6*16^{-1}+3*16^{-2}+13*16^{-3}=5843{,}388_{(10)}$.

ФОРМАТЫ ДАННЫХ В ЭВМ

Любая информация (числа, команды, аналого-цифровые записи и др.) представляются в ЭВМ в виде двоичных кодов фиксированной или переменной длины – двоичных слов. Отдельные элементы двоичного кода называются разрядами или битами (0,1). Современные ЭВМ имеют байт-ориентированную адресацию памяти: 1 байт = 8 бит. Наибольшее распространение получили ЭВМ, имеющие длину разрядной сетки в 4 байта или 32 двоичных разряда.

Известны две формы представления чисел - с фиксированной запятой (ФЗ) и плавающей запятой (ПЗ).

Двоичные операнды в форме с ФЗ имеют вид **целых чисел в дополнительном коде,** у которых крайний левый разряд - знаковый.

Двоичные числа с ПЗ изображаются по-разному в ЕС ЭВМ и ПЭВМ. Общим в изображении является лишь то, что порядки чисел имеют смещения. В ***ЕС ЭВМ*** смещенный порядок занимает семь разрядов (смещение=64) и размещается в старшем байте вместе со знаковым разрядом числа, остальные разряды занимает мантисса, изображаемая в 16СС. Каждые 4 разряда мантиссы воспринимаются ЭВМ как шестнадцатеричная цифра, а порядок показывает положение запятой в шестнадцатеричной мантиссе. Мантисса изображается в прямом коде и должна быть нормализована.

В ПЭВМ смещенный порядок занимает восемь разрядов (смещение=128), крайний левый разряд сетки отводится под знак числа, остальные разряды - под мантиссу, изображаемую в 2СС. Смещенный порядок содержит информацию о положении запятой в двоичной мантиссе числа. Для повышения точности представления мантиссы её старший разряд, который в нормализованной мантиссе всегда равен «1», не заносится в разрядную сетку, а просто подразумевается.

Сравнение представления чисел в ПЭВМ и ЕС ЭВМ в форме с ПЗ показывает существенное расширение диапазона представления чисел в ЕС ЭВМ, при изображении мантиссы числа в 16СС.

Пример 2.

Отрицательное число А=-5843,39 представить:

-в форме с ФЗ в 32-разрядной сетке;

-в форме с ПЗ в 32-разрядной сетке ЕС ЭВМ и ПЭВМ.

В предыдущем примере получено двоичное изображение числа:

$А=-1011011010011,011000111101_{(2)}$.

1. Число в форме с ФЗ – целое, в ДК; в свободные разряды –«1»

$=2^{-12}$

2. Число в форме с ПЗ в ПЭВМ

мантисса двоичная

3. Число в форме с ПЗ в ЕС ЭВМ

мантисса шестнадцате-

ричная

СЛОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ

Отрицательные числа в ЦВМ представлены в специальных кодах: прямом, обратном и дополнительном.

Прямой код (ПК) представляет абсолютное значение числа с закодированным знаком: «+» – «0», «-» - «1».

Обратный код (ОК) положительного числа совпадает с его прямым кодом, а для отрицательного числа в знаковый разряд заносится «1», в остальных разрядах цифры заменяются на взаимнообратные (0-1, 1-0), т.е. формируется поразрядное дополнение числа до единицы.

Дополнительный код (ДК) положительного числа совпадает с его прямым кодом, а для отрицательного числа в знаковый разряд заносится «1», в цифровой части числа цифры заменяются на взаимообратные и к полученному инверсному изображению прибавляется единица к младшему разряду, т.е. код является дополнением до основания СС.

Таким образом, положительные числа во всех кодах одинаковы, а отрицательные – различны.

Модифицированные обратный и дополнительный коды (МОК и МДК) имеют для изображения знака два соседних разряда: «+» – «00», «-» - «11». Эти коды используются для обнаружения признаков ПРС – переполнения разрядной сетки. ПРС возникает при сложении чисел с ФЗ одинакового знака, когда результат операции выходит за верхнюю границу диапазона представления чисел. Это приводит к потере старших разрядов числа, что недопустимо.

Формальным признаком ПРС при использовании МОК и МДК является появление запрещенных комбинаций в знаковых разрядах – «01» или «10».

Для исправления результата можно либо увеличить масштаб исходных операндов и выполнить операцию снова; либо увеличить масштаб результата, сдвинуть число вправо на один разряд, а в освободившийся старший знаковый разряд поместить значение из младшего знакового разряда.

Сложение чисел в форме с ФЗ в ОК: при алгебраическом сложении чисел в ОК со знаковым разрядом оперируют как с разрядом цифровой части числа, а при возникновении единицы переноса из знакового разряда ее прибавляют к младшему разряду числа.

Сложение чисел в форме с ФЗ в ДК: при алгебраическом сложении чисел в ДК результат получают также в ДК, а при возникновении единицы переноса из знакового разряда ее отбрасывают.

Сложение чисел в форме с ПЗ выполняется в несколько этапов. Любое число в форме с ПЗ представлено в разрядной сетке мантиссой и порядком:

$$A = \pm m_A \times 2^{\pm p_A}; \quad B = \pm m_B \times 2^{\pm p_B}.$$

Чтобы сложить два числа, надо выполнить различные действия над мантиссами и порядками. Поэтому в машинах предусмотрены различные устройства для обработки мантисс и порядков. Мантиссы исходных операндов нормализованы. Алгоритм сложения чисел с ПЗ.

1. Выравнивание порядков слагаемых: меньший порядок увеличивается до большего, при этом мантисса меньшего преобразуемого числа денормализуется.

В машине выполняется вычитание порядков операндов. Знак и модуль разности порядков определяет, мантиссу какого из слагаемых надо сдвигать вправо и на сколько разрядов.

2. Сложение мантисс операндов по правилам сложения чисел с ФЗ.

3. Нормализация мантиссы результата, если необходимо. При этом **денормализация вправо**, когда в старшем разряде двоичной мантиссы стоит «0», требует сдвига мантиссы влево и уменьшения порядка на соответствующее количество единиц. **Денормализация влево** означает временное ПРС мантиссы суммы, но в отличие от чисел с ФЗ здесь возможна коррекция: сдвиг мантиссы на один разряд вправо и увеличение на единицу порядка суммы.

При больших величинах порядков возможно истинное переполнение разрядной сетки со стороны порядков чисел с ПЗ, когда величина порядка оказывается настолько большой, что не может быть помещена в отводимые под порядок разряды. Однако, вероятность этого невелика.

Смещенные порядки используются в большинстве современных ЭВМ для упрощения процесса выравнивания порядков, их сравнения и ускорения выполнения различных операций.

При этом для представления порядка применяется специальный дополнительный код с инверсным кодированием знака: «+» – «1», «-» - «0». В результате порядки чисел увеличиваются (в ЕСЭВМ на 2^6=64, в СМЭВМ на 2^7=128), что приводит к смещению всех порядков по числовой оси в положительном направлении. Такие смещенные порядки называют **характеристиками**, а так как все характеристики - целые положительные числа, то алгебраическое сложение их можно выполнять без предварительного анализа знаков.

Например, изобразить в 7-миразрядной сетке ЕС ЭВМ характеристику, соответствующую порядку =(-26), и проверить смещение порядка.

$26_{(10)}=11010_{(2)}$ Порядок $=(-26)_{ПК}=\underline{1}.011010$

$(-26)_{ДК}=\underline{1}.100110$

Характеристика$=\underline{0}.100110$

Смещение порядка=64-26=38=100110

Пример 3. Сложить числа $A=30=11110_{(2)}$ и $B=72=1001000_{(2)}$, меняя знаки и форму представления.

а) Операнды отрицательны, сложить их в ОК в форме с ФЗ. $M=2^7$.

$A_{ПК}=1,0011110$ | $A_{ОК}=1,1100001$

$B_{ПК}=1,1001000$ | $B_{ОК}=1,0110111$

$(A+B)_{ОК}=1,0011001$ | $11,0011000$

$(A+B)_{ПК}=1,1100110$ $(M=2^7)$ | 1

$A+B=-1100110_{(2)}=-102_{(10)}$ | $(A+B)_{ОК}=1,0011001$

б) Знаки операндов $A<0$, $B>0$. Представить их в разрядной сетке условной машины в форме с ПЗ, при сложении мантисс использовать ДК.

Под мантиссы со знаком отведено восемь разрядов, под порядки со знаком - четыре разряда.

Мантисса	Порядок
A=1,1111000	0101
B=0,1001000	0111

1. Выравнивание порядков слагаемых, для чего выполняется их вычитание с использованием ДК:

$P_A=0101$

$P_B=1001$

$(P_A-P_B)_{ДК}=1110$ $(P_A-P_B)_{ПК}=1010$

Разность порядков =(–2), следовательно, мантиссу числа A надо сдвинуть на 2 разряда вправо и соответственно увеличить порядок на 2:

Мантисса	Порядок
A=1,0011110	0111

2. Сложение мантисс как чисел с ФЗ в ДК:

$1,1100010\ =m_A$

$0,1001000\ =m_B$

$\cancel{1}0,0101010\ = m_{A+}m_B$

3. Нормализация мантиссы результата путём сдвига на один разряд влево с одновременным уменьшением порядка на единицу.

Мантисса	Порядок
0,1010100	0110

Проверка: $0,1010100*2^6=42$.

Умножение двоичных чисел

Процесс умножения чисел в двоичной системе счисления прост, так как разрядами множителя могут быть либо «0», либо «1», следовательно, частичным произведением в каждом такте цикла умножения будет либо «0», либо множимое. Поэтому в цикле умножения двоичных чисел три элементарных операции:

- анализ цифры очередного разряда множителя;
- суммирование множимого с накапливаемой суммой частичных произведений, если цифра множителя =1;
- сдвиги в каждом такте умножения.

Умножение можно выполнять как с младших, так и со старших разрядов множителя, и сдвигать можно как сумму частичных произведений, так и множимое. Это и формирует четыре способа умножения чисел, схемы которых приведены на рис.1.

Следует обратить внимание на то, что множитель сдвигается во всех способах умножения, так как в каждом такте анализируется очередной разряд: при умножении с младших разрядов сдвиг выполняется вправо - в сторону младших разрядов, при умножении со старших разрядов множитель сдвигается влево. И еще одна особенность, позволяющая легко запомнить способы умножения: сумма частичных произведений всегда сдвигается в ту же сторону, что и множитель, а множимое сдвигается навстречу множителю, т.е. в противоположную сторону.

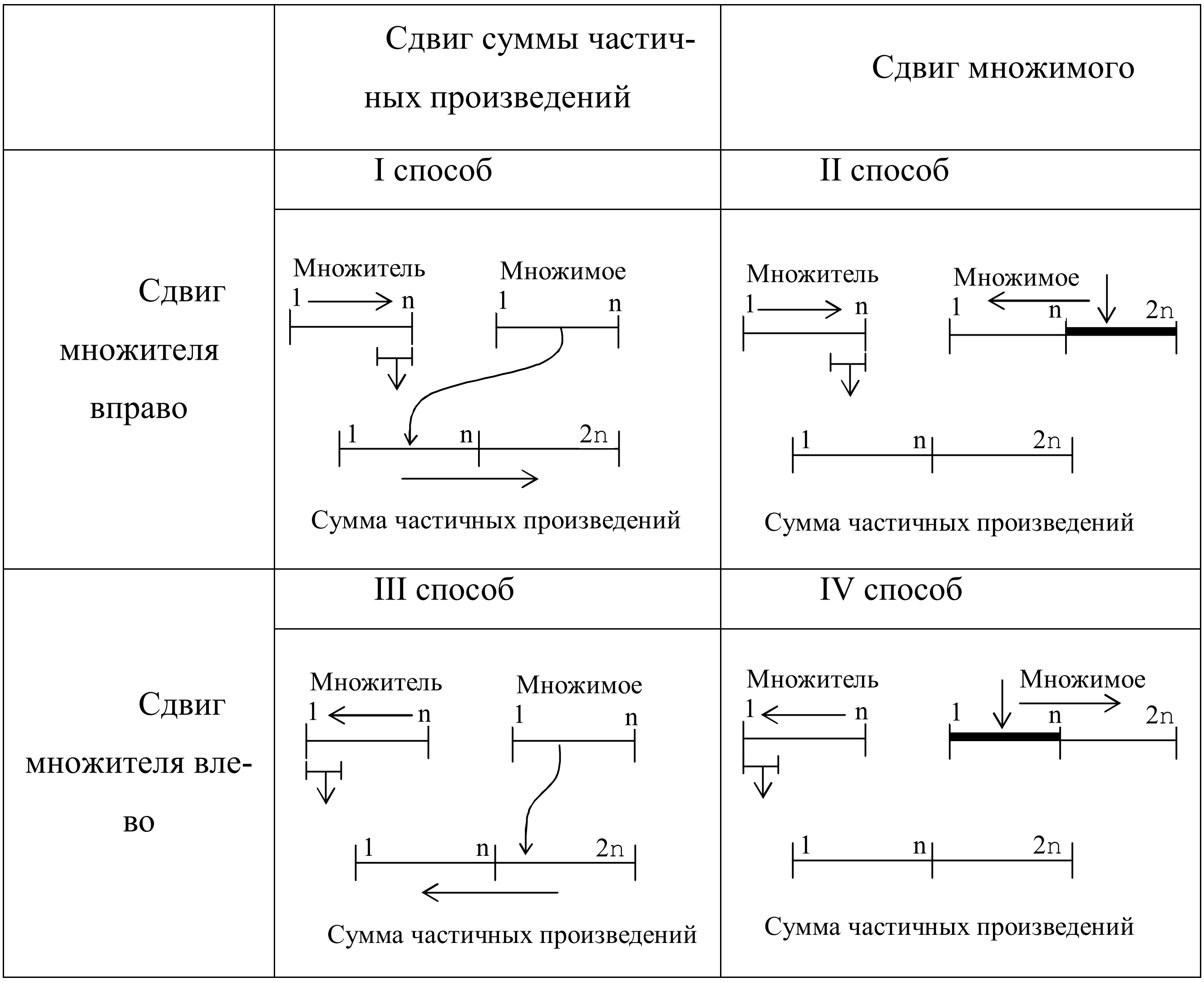


Рис.1. Схемы четырех способов умножения чисел

<u>*I способ – умножение с младших разрядов множителя со сдвигом суммы частичных произведений вправо*</u>

Устройства для хранения операндов - регистры, имеют следующую разрядность: регистры множителя и множимого - n-разрядные; регистр суммы частичных произведений - 2n-разрядный.

На схеме показано, что множимое следует прибавлять в старшие n разрядов регистра суммы частичных произведений. Причем разрядность регистра сумм можно уменьшить вдвое, до n-разрядов, помещая при сдвиге младшие разряды суммы на место освобождающихся разрядов регистра множителя.

Особенность I способа - в цикле умножения возможно ***временное переполнение разрядной сетки*** (ПРС) в регистре суммы частичных произведений, которое ликвидируется при очередном сдвиге вправо.

II способ – умножение с младших разрядов множителя со сдвигом множимого влево

Этот способ требует n–разрядного регистра множителя и двух 2n-разрядных регистров множимого и суммы частичных произведений. Причем, первоначально множимое помещается в младшие разряды регистра, а затем в каждом такте сдвигается на один разряд влево.

III способ – умножение со старших разрядов множителя со сдвигом суммы частичных произведений влево

Этот способ требует двух n–разрядных регистров множителя и множимого и одного 2n-разрядного регистра суммы частичных произведений. На схеме видно, что суммирование множимого следует выполнять в младшие n разрядов регистра суммы частичных произведений.

Особенность III способа - в последнем такте ***не следует выполнять сдвиг*** в регистре суммы частичных произведений.

IV способ – умножение со старших разрядов множителя со сдвигом множимого вправо

Этот способ требует одного n–разрядного регистра множителя и двух 2n-разрядных регистров множимого и суммы частичных произведений. Причем первоначально множимое помещается в старшие разряды регистра, а затем в каждом такте сдвигается на один разряд вправо.

Особенность IV способа - перед началом цикла умножения следует ***множимое сдвинуть*** на один разряд вправо.

Все приведенные выше четыре способа используются как в алгоритмах умножения в прямом коде (ПК), так и в алгоритмах умножения в дополнительном коде (ДК).

Рассмотрим алгоритмы умножения дробных чисел с фиксированной запятой (ФЗ).

Умножение чисел в прямом коде

Алгоритм умножения двоичных чисел в ПК.

1. Определить знак произведения путем сложения по модулю два знаковых разрядов сомножителей.
2. Перемножить модули сомножителей одним из четырех способов.
3. Присвоить полученному произведению знак из п.1.

Пример 4. Перемножить числа $A=26_{(10)}=11010_{(2)}$; $B=-19_{(10)}=-10011_{(2)}$, представив их в 2СС, ПК, с ФЗ.

<table>
<tr><td rowspan="2">M=2^5</td><td>A=0,1101
0</td></tr>
<tr><td>B=1,1001
1</td></tr>
</table>

1. Знак произведения: $0 \oplus 1 = 1$.
2. Перемножим модули сомножителей, используя I способ.

B=0,10011 – модуль множимого.

Таблица

$\longrightarrow$ Множитель	$\longrightarrow$ Сумма ЧП	Пояснения
0,1101**0**	0,0000000000	
0,_0_110**1**	0,0000000000	Сдвиги
	0,10011	
	0,1001100000	Сложение
0,_00_11**0**	0,0100110000	Сдвиги
0,_000_1**1**	0,0010011000	Сдвиги
	0,10011	
	0,1011111000	Сложение
	0,0101111100	Сдвиги
0,_0000_**1**	0,10011	
	0,1111011100	Сложение

Множитель	Сумма ЧП	Пояснения
0,*00000*	0,0111101110	Сдвиги

3. Прямой код произведения:

$$A \cdot B = 1{,}0111101110.$$

4. Проверка результата.

Масштаб произведения $M_A \cdot M_B = 2^{10}$.

$A \cdot B = -111101110_{(2)} = -494_{(10)}$.

Умножение чисел в дополнительном коде с простой коррекцией

Алгоритм умножения двоичных чисел в ДК с простой коррекцией.

1. Определить знак произведения путем сложения по модулю два знаковых разрядов сомножителей.
2. Перемножить модули сомножителей, представленных в ДК, одним из четырех способов – получить псевдопроизведение.
3. Если хотя бы один из сомножителей отрицателен, выполнить коррекцию по следующим правилам:

-если один сомножитель отрицателен, к псевдопроизведению прибавляется дополнительный код от модуля положительного сомножителя;

-если оба сомножителя отрицательны, к псевдопроизаведению прибавляются дополнительные коды от модулей дополнительных кодов обоих сомножителей, т.е. их прямые коды.

4. Присвоить модулю произведения знак из п.1 данного алгоритма.

Пример 5. Перемножить числа $A=-18_{(10)}=-10010_{(2)}$; $B=27_{(10)}=11011_{(2)}$, представив их в ДК и применив алгоритм с простой коррекцией.

$A_{ПК}=1{,}10010;$ $\qquad A_{ДК}=1{,}01110$

$M=2^5$

$B_{ПК}=0{,}11011;$ $\qquad B_{ДК}=0{,}11011$

1. Знак произведения: $1 \oplus 0 = 1$.

2. Перемножить модули сомножителей, используя II способ.

Таблица

→ Множитель	← Множимое	Сумма ЧП	Пояснения
0,0111**0**	0,0000011011	0,0000000000	Сдвиги
0,*0*011**1**	0,0000110110	0,0000110110	
0,*00*01**1**	0,0001101100	0,0000110110	Сложение
		0,0001101100	Сдвиги
		0,0010100010	Сложение
0,*000*0**1**	0,0011011000	0,0011011000	Сдвиги
0,*0000***0**	0,011011000	0,0101111010	Сложение
			Сдвиги
0,*00000*	0,1101100000		Сдвиги

Получено псевдопроизведение: 0,0101111010

3. Так как один из сомножителей отрицателен, нужна коррекция дополнительным кодом от модуля положительного сомножителя:

$B_{ДК}$=0,00101

$$\begin{array}{r} 0{,}0101111010 \\ 0{,}00101 \\ \hline 0{,}1000011010 \end{array}$$

Модуль полного произведения 0,1000011010.

4. Полное произведение в дополнительном коде:

$(A\cdot B)_{ДК}=1{,}1000011010.$

5. Проверка результата.

Масштаб произведения $M_A\cdot M_B=2^{10}$.

$(A\cdot B)_{ПК}=1{,}0111100110\quad (x2^{10})$.

$A \cdot B = -111100110_{(2)} = -486_{(10)}$.

Умножение чисел в дополнительном коде с автоматической коррекцией

Этот алгоритм разработан Бутом и является универсальным для умножения чисел в ДК. Сомножители участвуют в операции со знаковыми разрядами, которые рассматриваются как цифровые разряды числа. Результат получается сразу в дополнительном коде со знаком.

В процессе умножения анализируются две смежные цифры множителя: та, на которую выполняется умножение в данном такте, – m_1 и соседняя младшая цифра – m_2. В двоичном множителе этой паре цифр «m_1m_2» соответствуют четыре возможных набора – «00», «01», «10», «11», каждый из которых требует выполнения следующих действий:

1) набор **«01»** требует ***сложения*** множимого с предыдущей суммой частичных произведений;

2) набор **«10»** требует ***вычитания*** множимого из предыдущей суммы частичных произведений;

3) наборы **«00»** и **«11»** не требуют ***ни сложения, ни вычитания***, так как частичное произведение равно нулю.

В цикле умножения в каждом такте выполняются соответствующие сдвиги на один разряд. При этом могут использоваться все четыре способа умножения с некоторыми ***особенностями:***

- в I способе не следует выполнять последний сдвиг суммы частичных произведений;
- в IV способе не выполняется первый сдвиг множимого.

Это объясняется тем, что в этих тактах реализуется умножение не на цифровой, а на знаковый разряд числа.

Кроме того, при выполнении алгоритма умножения с автоматической коррекцией следует помнить ***о правилах сдвига отрицательных чисел в ДК***: при сдвиге ***влево*** освобождающиеся младшие разряды заполняются ***нулями***, при сдвиге ***вправо*** освобождающиеся старшие разряды заполняются ***единицами***, т.е. реализуется арифметический сдвиг числа.

Пример 6. Перемножить числа A=-$18_{(10)}$=-$10010_{(2)}$; B=$27_{(10)}$=$11011_{(2)}$, представив их в ДК и применив алгоритм с автоматической коррекцией. Выполнить умножение I и IV способами.

$M=2^5$ $A_{ПК}$ =1,10010; $A_{ДК}$ =1,01110 - множимое

      $B_{ПК}$ =0,11011; $B_{ДК}$ =0,11011 - множитель

Таблица

IV способ умножения

← Множитель	→ Множимое	Сумма ЧП	Пояснения
		0,0000000000	
0,11011	1,0111000000	1,0111000000	
		1,0111000000	Сложение
1,1011*0*	1,1011100000		Сдвиги
1,011*00*	1,1101110000	1,0111000000	Сдвиги
		0,0010010000	
		1,1001010000	Вычитание
0,11*000*	1,1110111000	1,1001010000	Сдвиги
		1,1110111000	
		~~1~~1,1000001000	Сложение
1,1*0000*	1,1111011100		Сдвиги
1,0*0000*	1,1111101110	1,1000001000	Сдвиги
		0,0000010010	
0,*00000*	1,111111011	1,1000010010	Вычитание
			Сдвиги

Получено произведение в дополнительном коде:

$(A\cdot B)_{ДК}$=1,1000011010.

Проверка результата.

Масштаб произведения $M_A \cdot M_B = 2^{10}$.

$(A \cdot B)_{ПК}=1{,}0111100110$ $(x2^{10})$.

$A \cdot B=-111100110_{(2)}=-486_{(10)}$.

Выполним умножение I способом, чтобы обратить внимание на необходимость сохранения предыдущей цифры множителя при сдвиге его вправо (в первом такте соседней младшей цифрой всегда является «0»).Аналогично следует поступать при умножении II способом.

Таблица

I способ умножения

Множитель →	Сумма ЧП →	Пояснения
0,1101**10**	0,0000000000	
	0,10010	
	0,1001000000	Вычитание
0,*0*1101**1**	0,0100100000	Сдвиги
0,*00*110**1**	0,0010010000	Сдвиги
	1,01110	
	1,1001010000	Сложение
0,*000*11**0**	1,1100101000	Сдвиги
	0,10010	
	~~1~~0,0101101000	Вычитание
0,*000*1**1**	0,0010110100	Сдвиги
0,*0000***1**	0,0001011010	Сдвиги
	1,01110	
	1,1000011010	Сложение
		Нет последнего сдвига!

Получено произведение в дополнительном коде:

$(A \cdot B)_{ДК}=1{,}1000011010$.

Проверка результата.

Масштаб произведения $M_A \cdot M_B = 2^{10}$.

$(A \cdot B)_{ПК} = 1,0111100110$ $(x2^{10})$.

$A \cdot B = -111100110_{(2)} = -486_{(10)}$.

Умножение чисел в форме с пЛавающей запятой

Когда сомножители заданы в форме с ПЗ

$$A = \pm m_A \cdot 2^{\pm p_A} \quad , \quad B = m_B \cdot 2^{\pm p_B},$$

то их произведение определяется следующим образом:

$$C = A \times B = \pm m_A \times m_B \times 2^{\pm(p_A + p_B)} = \pm m_C \times 2^{\pm p_C},$$

т.е. мантисса произведения m_C равна произведению мантисс сомножителей, а порядок p_C – сумме порядков сомножителей.

Это позволяет сформулировать алгоритм умножения чисел в форме ПЗ.

1. Определить знак произведения путем сложения по модулю два знаковых разрядов сомножителей.

2. Перемножить модули мантисс сомножителей по правилам умножения дробных чисел с ФЗ.

3. Определить порядок произведения алгебраическим сложением порядков сомножителей   с использованием ***модифицированных*** дополнительного или обратного кодов для выявления возможной ситуации ПРС.

4. Нормализовать мантиссу результата и выполнить округление, если это необходимо.

Примечания.

1. Так как мантиссы исходных сомножителей нормализованы, то денормализация мантиссы произведения возможна только на один разряд.

2. При умножении чисел с ПЗ возможно возникновении ПРС при сложении порядков, поэтому необходимо предусматривать выявление признаков ПРС в устройствах умножения чисел с ПЗ.

Пример 7. Перемножить числа $A=26_{(10)}$ и $B=-19_{(10)}$, представив их в форме с ПЗ в разрядной сетке условной машины. При умножении мантисс использовать III способ умножения.

$A=11010_{(2)}$.

$B=-10011_{(2)}$.

Операнды в разрядной сетке условной машины

0	11010	0	0101
1	10011	0	0101
знак числа	Мантисса пять разрядов	знак порядка	Порядок четыре разряда

1. Знак произведения: $0\oplus1=1$.
2. Произведение модулей мантисс.

Таблица

← Множитель	← Сумма ЧП	Пояснения
	0,0000000000	
0,**1**101*0*	10011	
	0,0000010011	Сложение
0,**1**01*00*	0,0000100110	Сдвиги
	10011	
	0,0000111001	Сложение
0,**0**1*000*	0,0001110010	Сдвиги
	0,0011100100	Сдвиги
0,**1**_0000_	10011	
	0,0011110111	Сложение
0,**0**_0000_	0,0111101110	Сдвиги
		Нет последнего сдвига!

Полное 10-разрядное произведение модулей мантисс: 0,0111101110.

3. Порядок произведения

0 0101
0 0101

0 1010.

4. Нормализация и округление мантиссы произведения.

Так как в разрядной сетке условной машины под мантиссу отведено 5 разрядов, то необходимо округлить мантиссу результата, что приводит к погрешности. Для уменьшения погрешности следует ***сначала выполнить нормализацию*** мантиссы произведения, чтобы больше верных цифр попало в разрядную сетку, а затем округлить мантиссу. Обычно выполняют ***симметричное округление***: если первый отбрасываемый разряд = 1, то к младшему разряду мантиссы в разрядной сетке следует прибавить единицу; если отбрасываемый разряд = 0, мантисса остаётся без изменения.

Результат в разрядной сетке

	11111		1001

Нормализация мантиссы произведения выполнятся сдвигом ее влево на один разряд с одновременным уменьшением порядка на единицу.

Проверка: $-0,11111 \cdot 2^9 = -111110000_{(2)} = -496_{(10)}$.

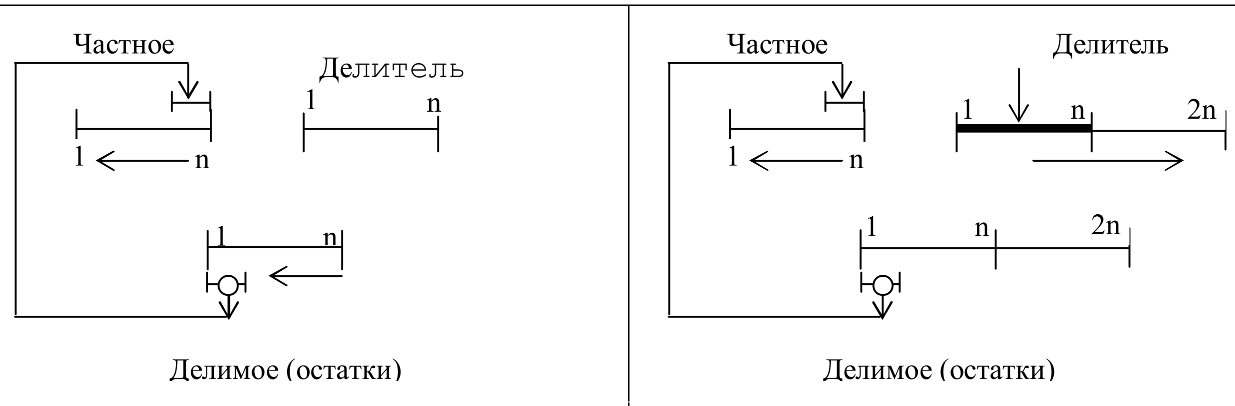
Абсолютная ошибка округления $= +2_{(10)}$.

ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ

Процесс деления состоит из последовательности операций вычитания и сдвигов, при этом операция вычитания заменяется операцией сложения остатка с делителем, представленным в обратном или дополнительном кодах.

При делении чисел в прямом коде знак частного определяется сложением по модулю два знаковых разрядов делимого и делителя, и далее в процессе деления участвуют модули операндов.

Так как операция деления обратна умножению и начинается всегда со старших разрядов, то существуют два способа деления – обращенный третий и четвертый способы умножения (рис. 2). Причем нередко для реализации умножения и деления целесообразно использовать одно и то же оборудование: регистр множимого как регистр делителя, регистр множителя - как регистр частного, а регистр частных сумм - как регистр делимого, в который затем заносят остатки от деления.



I способ (обращенный III способ умножения)

II способ (обращенный IV способ умножения)

Рис. 2. - Схемы способов деления чисел

Приведенные выше два способа деления можно выполнять, используя два алгоритма:

- с восстановлением остатков;
- без восстановления остатков.

Алгоритм деления с восстановлением остатков

В основе алгоритма деления лежит логика ручного счета. При выполнении деления на бумаге вычислитель быстро анализирует, что больше – делитель или делимое (очередной остаток), и когда делимое меньше делителя, в очередной разряд частного заноситься «0» и выполняется сдвиг.

В ЦВМ такой анализ можно сделать посредством вычитания делителя из делимого, и при получении отрицательного остатка в очередной разряд частного заносится «0», а отрицательный остаток следует восстановить до предшествующего значения, прибавив к нему делитель. Только после этого можно выполнить сдвиги. Если же остаток положителен, в частное заносится «1» и выполняются соответствующие способу деления сдвиги.

Это позволяет сформулировать ***алгоритм деления с восстановлением остатков*** для дробных чисел с ФЗ.

1. Определить знак частного сложением по модулю 2 знаковых разрядов делимого и делителя. Далее использовать модули операндов.

2. Вычесть из делимого делитель путем сложения в обратном или дополнительном кодах.

3. Проанализировать знак остатка после первого вычитания:

- если остаток положительный, произошло ПРС, операцию прекратить до смены масштабов операндов;

- если остаток отрицательный, в частное заносится «0» (этот разряд по окончании деления станет знаковым разрядом частного) и восстановить остаток, прибавив к нему делитель.

4. Выполнить сдвиги: частного на один разряд влево и остатка на один разряд влево (I способ) или делителя на один разряд вправо (II способ).

5. В цикле формирования цифр частного: вычесть из остатка делитель, прибавив его в обратном или дополнительном кодах.

6. Проанализировать знак полученного остатка:

- если остаток положителен, в частное занести «1»;
- если остаток отрицателен, в частное занести «0».

7. Восстановить отрицательный остаток, сложив его с делителем.

8. Выполнить сдвиги, как указано в пункте 4 алгоритма.

9. Завершить цикл формированием (n+1)–го остатка для округления. Последний сдвиг частного не выполнять.

10. Выполнить округление результата и присвоить частному знак, полученный в пункте 1 алгоритма.

В соответствии с вышеизложенным алгоритмом можно формально записать правила формирования очередного остатка для I и II способов деления.

Пусть D – делитель, Δ_i – остаток на i-м шаге алгоритма.

I способ деления требует сдвига влево на один разряд (удвоение) остатка (восстановленного - (Δ_i+D) или невосстановленного Δ_i):

$$\Delta_{I+1} = \begin{cases} 2\Delta_i - D, если \quad \Delta_i \geq 0, \\ 2(\Delta_i + D) - D, если \quad \Delta_i < 0 \end{cases} \qquad (1)$$

II способ деления требует сдвига вправо на один разряд делителя, т.е. уменьшения его вдвое:

$$\Delta_{I+1} = \begin{cases} \Delta_i - D/2, если \quad \Delta_i \geq 0, \\ (\Delta_i + D) - D/2, если \quad \Delta_i < 0 \end{cases} \qquad (2)$$

Анализ приведенного алгоритма позволяет отметить следующие недостатки:

- процесс деления ацикличен, так как операция восстановления остатка появляется нерегулярно, что приводит к усложнению устройства управления делением;
- быстродействие алгоритма невелико, т.к. примерно в половине шагов цикла выполняется дополнительная операция восстановления остатка.

Пример 8. Числа $A=27_{(10)}$ и $B=-30_{(10)}$ представить в форме с ФЗ в ПК ($M_{A,B}=2^5$) и разделить. A=0,11011 –делимое; B=1,11110-делитель.

1. Знак частного: 0⊕1=1.

2. Деление модулей операндов выполним I способом с использованием ДК при вычитании.

Таблица

← Частное	← Делимое (остатки)	Пояснения
0,00000	0,11011	Вычитание
	1,00010	
0,0000**<u>0</u>**	1,11101	Первый остаток
	0,11110	Восстановление
	0,11011	
	1,10110	Сдвиги
	1,00010	Вычитание
0,000**<u>01</u>**	0,11000	Второй остаток
	1,10000	Сдвиги
	1,000010	Вычитание
0,00**<u>011</u>**	0,10010	Третий остаток

← Частное	← Делимое (остатки)	Пояснения
	1,00100	Сдвиги
	1,00010	Вычитание
0,0**<u>0111</u>** ←	0,00110	Четвертый остаток
	0,01100	Сдвиги
	1,00010	Вычитание
0,**<u>01110</u>** ←	1,01110	Пятый остаток
	0,11110	Восстановление
	0,01100	
	0,11000	Сдвиги
	1,00010	Вычитание
<u>0,11100</u> ←	1,11010	Шестой остаток
	0,11110	Восстановление
	0,11000	
	1,10000	Сдвиг остатка
	1,00010	Вычитание
<u>0,11100</u>(1) ←	0,100010	Седьмой остаток для округления

1. Так как седьмой остаток положительный, то в отбрасываемый разряд частного должна быть занесена «1», следовательно, для округления результата к младшему разряду частного нужно прибавить единицу

Тогда модуль частного после округления:

$$A/B=0,11101.$$

2. Частное со знаком в прямом коде:$(A/B)_{ПК}$= 1,11101.

Проверка: A/B= $-0,11101_{(2)}$=$-0,90625_{(10)}$.

Точный результат –(27/30)=-0,9.

Для демонстрации ситуации ПРС при делении дробных чисел рассмотрим следующий пример.

Пример 9. Разделить А=-25(10) на В=9(10), операнды в форме с ФЗ в прямом коде ($M_{A,B}=2^5$).

А=1,11001 – делимое; В=0,01001 – делитель.

1. Знак частного: 1⊕0=1.

2. Деление модулей операндов выполним II способом с использованием ДК при вычитании.

Таблица

← Частное	Делитель →	Делимое (остатки)	Пояснения
	0,010010	0,1100100000	
	0000	1,1011100000	
		~~1~~0,1000000000	ПРС!
Увеличим масштаб делимого до $M=2^7$			
	0,010010	0,00110 01000	
	0000	1,10111 00000	Вычитание
0,0000**0** ←		1,11101 01000	Первый остаток
		0,01001 00000	Восстановление
		~~1~~0,00110 01000	
	0,001001	0,00110 01000	Сдвиги
	0000	1,11011 10000	Вычитание
0,000**01** ←		~~1~~0,00001 11000	Второй остаток
	0,000100	0,00001 11000	Сдвиги
	1000	1,11101 11000	Вычитание
0,00**010** ←		1,11111 10000	Третий остаток
		0,00010 01000	Восстановление
		~~1~~0,00001 11000	
	0,000010	0,00001 11000	Сдвиги
	0100	1,11110 11100	Вычитание
0,0**0101** ←	0,000001	~~1~~0,00000 10100	Четв-ый остаток
	0010	0,00000 10100	Сдвиги

← Частное	→ Делитель	Делимое (остатки)	Пояснения
		1,11111 01110	Вычитание
0,**01011** ←	0,000000 1001	~~1~~0,00000 00010	Пятый остаток
		0,00000 00010	Сдвиги
		1,11111 10111	Вычитание
0,10110 ←		1,11111 11001	Шестой остаток
		0,00000 01001	Восстановление
	0,000000 0100	0,00000 00010	
		0,00000 00010	Сдвиги
		1,11111 11100	Вычитание
0,10110(0) ←		1,11111 11110	Седьмой остаток

3. Округленное частное в прямом коде:

A/B=1,10110

Проверка: $M=M_A/M_B=2^7/2^5=2^2$.

A/B= $-10,110_{(2)}=-2,75_{(10)}$.

Точный результат: –(25/9)=-2,78; относительная погрешность =1,08%.

Алгоритм деления без восстановления остатков

Для исключения недостатков предыдущего алгоритма был предложен алгоритм деления без восстановления остатков, основанный на простейших преобразованиях приведенных ранее формул (1) и(2).

В I способе деления после упрощения второй строки формулы (1)

$$\Delta_{i+1} = \begin{cases} 2\Delta_i - D, если \quad \Delta_i \geq 0, \\ 2\Delta_i + D, если \quad \Delta_i < 0, \end{cases} \qquad (3)$$

т.е. вместо восстановления отрицательного остатка следует удваивать любой остаток сдвигом его на один разряд влево и складывать делитель с остатком, если остаток отрицательный, или вычитать делитель из остатка, если остаток положительный.

Во II способе деления после упрощения второй строки формулы (2)

$$\Delta_{i+1} = \begin{cases} \Delta_i - D/2, если \quad \Delta_i \ge 0, \\ \Delta_i + D/2, если \quad \Delta_i < 0, \end{cases} \qquad (4)$$

т.е. в каждом такте цикла деления следует уменьшать вдвое делитель сдвигом его на один разряд вправо и складывать его с остатком, если остаток отрицателен, или вычитать делитель из остатка, если остаток положителен.

Это позволяет сформулировать ***алгоритм деления без восстановления остатков*** для дробных чисел с ФЗ.

1. Определить знак частного сложением по модулю два знаковых разрядов делимого и делителя. Далее использовать модули операндов.

2. Вычесть из делимого делитель путем сложения в обратном или дополнительном кодах.

3. Проанализировать знак остатка после первого вычитания:

- если остаток положительный, произошло ПРС, операцию следует прекратить для смены масштабов операндов;

- если остаток отрицательный, в частное занести «0» и продолжить операцию деления.

4. Выполнить сдвиги частного на один разряд влево и остатка на один разряд влево (I способ) или делителя на один разряд вправо (II способ).

5. Если до сдвига остаток был положительным, вычесть из остатка делитель, если остаток был отрицательным, прибавить к остатку делитель.

6. Если вновь полученный остаток положительный, в очередной разряд частного занести «1», в противном случае – «0».

7. Выполнить пункты 4-6 алгоритма (n+1) раз, причем, последний сдвиг частного не выполнять, так как (n+1) разряд - для округления.

8. Выполнить округление результата и присвоить частному знак из пункта 1 алгоритма.

Пример 10. Числа $A=-12_{(10)}$ и $B=-18_{(10)}$ представить в форме с ФЗ в прямом коде ($M_{A,B}=2^5$) и разделить, используя алгоритм без восстановления остатков,I способ деления и ОК при вычитании.

A=1,01100 – делимое; B=1,10010 – делитель.

1. Знак частного: $1\oplus1=0$.

2. Деление модулей операндов выполняется I способом.

Таблица

Частное	Делимое (остатки)	Пояснения
0,00000	0,01100	Вычитание
	1,01101	
	1,11001	Первый остаток<0
0,0000**0**	1,10011	Сдвиги
	0,10010	Сложение
	10,00101	
	1	
	0,00110	Второй остаток>0
0,000**01**	0,01100	Сдвиги
	1,01101	Вычитание
	1,11001	Третий остаток<0
0,00**010**	1,10011	Сдвиги
	0,10010	Сложение
	10,00101	
	1	
	0,00110	Четвертый остаток>0
0,0**0101**	0,01100	Сдвиги
	1,01101	Вычитание
	1,11001	Пятый остаток<0

Частное	Делимое (остатки)	Пояснения
0,**01010**	1,10011 0,10010	Сдвиги Сложение
	10,00101 1	
 0,**10101**	0,00110 0,01100 1,01101	Шестой остаток>0 Сдвиги Вычитание
0,**10101**(0)	1,11001	Седьмой остаток

3. Модуль частного после округления:

A/B=0,10101.

Проверка: A/B= $0,10101_{(2)}=0,656_{(10)}$.

Точный результат: (12/18)=0,667; относительная погрешность =1,65%.

Замечание. *Следует обратить внимание на особенности использования* ***ОК при вычитании:*** *при сдвиге* ***отрицательных чисел*** *как влево, так и вправо освобождающиеся разряды заполняются «1». Кроме того, в соответствии с правилами сложения чисел в ОК при возникновении единицы переноса из знакового разряда её следует прибавлять к младшему разряду числа.*

Алгоритм делениЯ в дополнительном коде

Так как числа с ФЗ представлены в современных ЭВМ в дополнительном коде, то целесообразно и операции над ними выполнять в дополнительном коде. Рассмотрим алгоритм деления в ДК с автоматической коррекцией (аналог алгоритма Бута).

Операнды участвуют в операции деления со знаковыми разрядами, и знак частного определяется в процессе деления.

1. Если знаки делимого и делителя совпадают, в частное заносится «0», в противном случае –«1». Этот разряд знаковый.

2. Если знаки операндов совпадают, делитель вычитается из делимого, в противном случае – делитель прибавляется в делимому.

3. Если знак первого остатка совпадает со знаком делимого, произошло ПРС, и операцию деления следует прекратить. В противном случае деление продолжить.

4. Выполнить сдвиги: частного и остатка на один разряд влево (I способ) или делителя на один разряд вправо (II способ).

5. Все последующие остатки формируются по следующему правилу:

- если знаки делителя и остатка ***до сдвига*** совпадают, делитель вычесть из остатка, в противном случае делитель прибавить к остатку.

6. Если знаки нового остатка и делителя совпадают, в очередной разряд частного занести «1», в противном случае – «0».

7. Выполнить пункты 4-6 алгоритма (n+1) раз с учетом формирования разряда частного для округления. Последний сдвиг частного не выполнять.

8. Выполнить округление результата.

Пример 11. Числа $A=-26_{(10)}$ и $B=29_{(10)}$ представить в форме с ФЗ в дополнительном коде ($M_{A,B}=2^5$) и разделить, используя алгоритм деления в дополнительном коде с автоматической коррекцией.

$A_{пк}$=1,11010 – делимое; $B_{пк}$=0,11101 – делитель.

Переводим операнды в дополнительный код:

$A_{дк}$=1,00110 – делимое; $B_{дк}$=0,11101 – делитель.

Деление операндов выполним I способом с использованием ДК при вычитании.

Таблица

← Частное	← Делимое (остатки)	Пояснения
0,0000**1**	1,00110 0,11101	Сравнение знаков Сложение
0,000**10**	10,00011 0,00110 1,00011	1-й ост.- нет ПРС Сдвиги, сравнение знаков В и 1-й ост. Вычитание
0,00**100**	1,01001 0,10010 0,11101	2-ой ост., в частное – 0 Сдвиги, сравнение знаков В и 2-й ост. Сложение
0,0**1000**	1,01111 0,11110 0,11101	3-й остаток, в частное – 0 Сдвиги, сравнение знаков В и 3-й ост. Сложение
0,**10001**	1,11011 1,10110 0,11101	4-й остаток, в частное – 0 Сдвиги, сравнение знаков В и 4-й ост. Сложение
1,00011	10,10011 1,00110 1,00011	5-й остаток, в частное – «1» Сдвиги, сравнение знаков В и 5-й ост. Вычитание
1,00011(0)	10,01001 0,10010 1,00011	6-ой остаток, в частное – «1» Сдвиг остатка Вычитание
	1,10101	7-й остаток, в частное – «1»

3. Частное после округления в дополнительном коде:

$$A/B_{дк}=1{,}00011.$$

Проверка. Частное в прямом коде: $A/B_{пк}=1{,}11101$;

$$A/B= -0{,}11101_{(2)}=-0{,}90625_{(10)}.$$

Точный результат:-(26/29)=-0,89655;относительная погрешность =1,08%.

Деление чисел в форме с плавающей запятой

Когда операнды заданы в форме с ПЗ:

$$A = \pm m_A \times 2^{\pm p_A} \qquad B = \pm m_B \times 2^{\pm p_B},$$

то их частное определяется следующим образом:

$$C = \frac{A}{B} = \pm \frac{m_A}{m_B} \times 2^{\pm(p_A - p_B)} = \pm m_C \times 2^{\pm p_C},$$

т.е. мантисса частного m_C есть частное от деления мантиссы делимого на мантиссу делителя, а порядок частного p_C есть разность порядков операндов. Это позволяет сформулировать алгоритм деления чисел в форме ПЗ.

1. Определить знак частного путем сложения по модулю два знаковых разрядов операндов.
2. Разделить модуль мантиссы делимого на модуль мантиссы делителя по правилам деления дробных чисел с ФЗ.
3. Определить порядок частного вычитанием порядка делителя из порядка делимого, используя при вычитании ОК или ДК.
4. Нормализовать мантиссу результата и присвоить знак, определенный в пункте 1 алгоритма.

В отличие от деления чисел в форме с ФЗ при делении чисел с ПЗ получение положительного остатка при первом вычитании не означает ПРС. Для чисел с ПЗ следует денормализовать мантиссу делимого сдвигом ее на один разряд вправо, увеличить на единицу порядок делимого и снова выполнить первое вычитание.

Однако, ситуация ПРС при делении чисел с ПЗ возможна при вычитании порядков операндов, если они были разных знаков.

Пример 12. Числа А=$26_{(10)}$ и В=$-19_{(10)}$ представить в форме с ПЗ в разрядной сетке условной машины и разделить, применив при делении модулей мантисс II способ с использованием ОК при вычитании и алгоритм деления без восстановления остатков.

A =11010$_{(2)}$

B =-10011$_{(2)}$

Операнды в разрядной сетке условной машины

0	11010	0	0101
1	10011	0	0101
знак числа	Мантисса пять разрядов	знак порядка	Порядок четыре разряда

1. Знак частного: 0⊕1=1.
2. Частное от деления модулей мантисс.

Таблица

← Частное	→ Делитель	Делимое (остатки)	Пояснения
	0,1001100000	0,11010 00000 1,01100 11111	1-е вычитание
		~~1~~0,00110 11111 1	
		0,00111 00000	1-й ост.>0 – признак ПРС!
	0,1001100000	0,01101 00000 1,01100 11111	Вычитание
		1,11001 11111	1-й ост.<0
0,0000**0**	0,0100110000	1,11001 11111	Сдвиги
		0,01001 10000	Сложение
		10,00011 01111 1	
		0,00011 10000	2-й ост.>0
0,000**01**	0,0010011000	0,00011 10000	Сдвиги
		1,11011 00111	Вычитание
		1,11110 10111	3-й ост.<0

Частное	Делитель	Делимое (остатки)	Пояснения
0,00**010**	0,0001001100	1,11110 10111	Сдвиги
		0,00010 01100	Сложение
		10,00001 00011	
		1	
		0,00001 00100	4-й ост.>0
0,0**0101**	0,0000100110	0,00001 00100	Сдвиги
		1,11110 11001	Вычитание
		1,11111 11101	5-й ост.<0
0,**01010**	0,0000010011	1,11111 11101	Сдвиги
		0,00000 10011	Сложение
		10,00000 10000	
		1	
		0,00000 10001	6-й ост.>0
0,10101	0,0000001001	0,00000 10001	Сдвиги
		1,11111 10110	Вычитание
		10,00000 00111	
		1	
0,10101(1)		0,00000 01000	7-й ост.>0
			Разряд для округления

Округление мантиссы частного:

0,10101

______1

0,10110

3.Вычитание порядков в ДК:

00110

11011

~~1~~00001 -порядок частного.

3. Мантисса нормализована. Результат в разрядной сетке:

	1 0110		001

Проверка: -0,10110*2^1=-1,0110$_{(2)}$=-1,375$_{(10)}$.

Точный результат: A/B=-(26/19)=-1,368$_{(10)}$; отн. погрешность = 0,51%.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Савельев А.И. Прикладная теория цифровых автоматов.- М: Высшая школа,1996. – 272 с.

2. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. – Минск: Выс.шк.,1980. –335 с.

3. Дудкин В.С., Кутепова Е.С., Матвеев В.Д. Машинные алгоритмы десятичной арифметики. – Горький: Изд-во ГГУ,1882. –59 с.

4. Ростовцев В.С., Блинова С.Д. Оформление курсовых и дипломных проектов для студентов специальности 230101 – Киров: Изд-во ВятГТУ, 2006.- 39с.