

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

Т.Р. ФАДЕЕВА, М.Л. ДОЛЖЕНКОВА

Организация арифметических операций над дво- ичными числами

Методические указания к курсовой работе

Дисциплина "Дискретная математика"

Для студентов специальности 220100

Киров 2001

УДК 681.332(07)

Ф152

Содержание

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ	4
Формирование операндов	4
Задание 1	4
Задание 2	4
Задание 3	5
Задание 4	6
Задание 5	6
Задание 6	6
ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ ПОЗИЦИОННОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ	7
ПЕРЕВОД ЦЕЛЫХ ЧИСЕЛ	7
ПЕРЕВОД ПРАВИЛЬНЫХ ДРОБЕЙ	7
ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ СИСТЕМ СЧИСЛЕНИЯ	8
ФОРМАТЫ ДАННЫХ В ЭВМ	9
СЛОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ	11
УМНОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ	14
УМНОЖЕНИЕ ЧИСЕЛ В ПРЯМОМ КОДЕ	17
УМНОЖЕНИЕ ЧИСЕЛ В ДОПОЛНИТЕЛЬНОМ КОДЕ С ПРОСТОЙ КОРРЕКЦИЕЙ	18
УМНОЖЕНИЕ ЧИСЕЛ В ДОПОЛНИТЕЛЬНОМ КОДЕ С АВТОМАТИЧЕСКОЙ КОРРЕКЦИЕЙ	20
УМНОЖЕНИЕ ЧИСЕЛ В ФОРМЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ	23
ДЕЛЕНИЕ ДВОИЧНЫХ ЧИСЕЛ	25
ОСОБЕННОСТИ ДЕЛЕНИЯ ДРОБНЫХ И ЦЕЛЫХ ЧИСЕЛ	26
АЛГОРИТМ ДЕЛЕНИЯ С ВОССТАНОВЛЕНИЕМ ОСТАТКОВ	27
АЛГОРИТМ ДЕЛЕНИЯ БЕЗ ВОССТАНОВЛЕНИЯ ОСТАТКОВ	31
АЛГОРИТМ ДЕЛЕНИЯ В ДОПОЛНИТЕЛЬНОМ КОДЕ	34
ДЕЛЕНИЕ ЧИСЕЛ В ФОРМЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ	36
КАЛЕНДАРНЫЙ ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ	40
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	40

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Формирование операндов

Для выполнения алгоритмов различных арифметических операций каждый студент должен сформировать две пары чисел по следующим правилам:

Числа А и В – смешанные десятичные числа, содержащие три значащих цифры в целой части и две значащие цифры в дробной части; причем одно число следует взять из интервала $[260;500]$, второе – из $[600;900]$.

Числа С и D – целые двухразрядные десятичные числа из интервала $[20;90]$, исключая числа 32 и 64 и их ближайшую окрестность, чтобы их двоичное изображение не содержало преимущественно «1» или «0».

Выбранные операнды А,В,С,D следует согласовать с преподавателем.

Задание 1

Перевод чисел из одной позиционной системы в другую с использованием промежуточных систем счисления и изображение чисел в формате современных ЦВМ.

1. Числа А и В перевести из 10СС в 2СС, используя 8СС и 16СС в качестве промежуточных, а затем выполнить проверку правильности перевода. Число А: $10СС \rightarrow 8СС \rightarrow 2СС \rightarrow 16СС \rightarrow 10СС$. Число В: $10СС \rightarrow 16СС \rightarrow 2СС \rightarrow 8СС \rightarrow 10СС$.

2. Пусть $A > 0$, $B < 0$. Изобразить каждое число в форме с ФЗ в 32-разрядной сетке ЦВМ, указав масштаб операндов.

3. Пусть $A < 0$, $B > 0$. Изобразить каждое число в форме с ПЗ в 32-разрядной сетке ЦВМ, представив мантиссу в 2СС (ПЭВМ) и 16СС (ЕС ЭВМ) и отведя соответственно под смещенные порядки (характеристики) восемь разрядов (ПЭВМ) и семь разрядов (ЕС ЭВМ).

Задание 2

Сложение двоичных чисел. Выполнить сложение А и В, изменяя их знаки, форму представления и используя различные коды.

1. Знаки операндов: $A > 0$, $B < 0$. Сложить числа с ФЗ в обратном коде. Проверить результат операции.
2. Знаки операндов: $A < 0$, $B > 0$. Сложить числа с ФЗ в дополнительном коде. Проверить результат операции.
3. Оба операнда отрицательные. Сложить числа в форме с ФЗ в одном из модифицированных кодов – МОК или МДК. При возникновении ситуации ПРС выполнить корректирующие действия и проверить результат.
4. Оба операнда положительные. Сложить числа в форме с ПЗ. Для этого сначала исходные данные изобразить в разрядной сетке условной машины, выбрав необходимое количество разрядов для изображения мантиссы со знаком и порядка со знаком. При этом помнить, что мантиссы исходных операндов должны быть нормализованы. Затем последовательно выполнить все пункты алгоритма сложения чисел с ПЗ, получить результат операции, изобразить его в разрядной сетке той же условной машины и выполнить проверку результата.

Задание 3

Умножение двоичных чисел. Числа C и D перевести в 2 СС и перемножить, изменяя их знаки и форму представления, используя различные алгоритмы и способы умножения.

1. Знаки операндов: $C > 0$, $D < 0$. Умножить числа с ФЗ в прямом коде, используя первый способ умножения. Проверить результат операции.
2. Знаки операндов: $C < 0$, $D > 0$. Перемножить числа с ФЗ в дополнительном коде, используя II способ и алгоритм с автоматической коррекцией. Проверить результат операции.
3. Оба операнда отрицательные. Представить их в форме с ФЗ в дополнительном коде и перемножить их III способом, используя алгоритм с простой коррекцией. Проверить результат.
4. Оба операнда положительные. Представить числа в форме с ПЗ. Для этого сначала исходные данные изобразить в разрядной сетке условной машины (с порядками). При умножении мантисс использовать IV способ умножения. Последовательно выполнить все пункты алгоритма умножения чисел в форме с ПЗ, изобразить результат в разрядной сетке выбранной условной машины и выполнить проверку результата.

Задание 4

Деление двоичных чисел.

1. Знаки операндов: $C > 0$, $D < 0$, причем C – делимое. Представить числа в форме с ФЗ в прямом коде, выполнить деление первым способом, применив алгоритм деления с восстановлением остатков с использованием ОК при вычитании. Проверить результат операции, оценить погрешность округления.

2. Оба операнда отрицательны, C – делимое. Выполнить операцию деления II способом чисел в форме с ФЗ в ПК, применив алгоритм деления без восстановления остатков с использованием ДК при вычитании.

3. Знаки операндов: $C < 0$, $D > 0$, причем D – делимое. Представить числа в форме с ФЗ в ДК, выполнить деление II способом, в соответствии с алгоритмом деления в ДК (с автоматической коррекцией) Проверить результат операции, оценить погрешность округления.

4. Оба операнда положительны, D – делимое. Представить числа в форме с ПЗ в разрядной сетке условной машины. Разделить числа, используя I способ деления, алгоритм выбрать самостоятельно. Изобразить частное в разрядной сетке условной машины и проверить результат операции.

Задание 5

Сложение двоично-десятичных чисел. Сложить смешанные числа A и B , изменяя их знаки и представляя их в различных двоично-десятичных кодах в форме с ФЗ.

Задание 6

Умножение двоично-десятичных чисел. В качестве операндов взять целую часть чисел A и B , представив их в кодах с естественными весами, перемножить, используя два алгоритма [3]:

- старорусский метод удвоения – деления пополам;
- метод десятично-двоичного разложения множителя.

Результат проверить.

ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ ПОЗИЦИОННОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

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

$$A(q) = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m} = \sum_{k=-m}^n a_k q^k,$$

где a_k – цифра числа в данной СС;

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 старой СС, q – основание новой СС, в которую надо перевести число $A(p)$. Тогда по основной формуле его можно представить в СС с новым основанием q :

$$A(q) = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0,$$

где $a_i < q$ – цифры числа в СС с новым основанием.

Разделим обе части приведенной формулы на новое основание q :

$$A(q)/q = a_n q^{n-1} + a_{n-1} q^{n-2} + \dots + a_1 + a_0/q.$$

Таким образом, младшая цифра a_0 числа $A(p)$ в новой СС определилась как остаток от деления исходного числа на новое основание q .

Далее целую часть первого частного снова делят на 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 нового числа.

ПЕРЕВОД ПРАВИЛЬНЫХ ДРОБЕЙ

Пусть по-прежнему p – старое основание, q – новое основание СС. Запишем правильную дробь в СС с новым основанием q :

$$A(q) = a_{-1}q^{-1} + a_{-2}q^{-2} + \dots + a_{-m}q^{-m}.$$

Умножим обе части равенства на новое основание q :

$$A(q) \cdot q = a_{-1} + a_{-2}q^{-1} + \dots + a_{-m}q^{-m+1}.$$

Так определилась первая после запятой цифра a_1 исходной дроби в новой СС. Далее следует умножить на новое основание q дробную часть первого произведения и определить вторую цифру новой дроби a_2 и т.д. Отсюда следует правило.

Чтобы перевести правильную дробь из одной позиционной СС в другую, надо ее 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 исходного числа, надо определить количество цифр в изображении числа по новому основанию.

Если m_1 – количество цифр исходной дроби с основанием p , m_2 – количество цифр по новому основанию q , то из условия сохранения точности $p^{-m_1} = q^{-m_2}$ можно получить выражение $m_2 = m_1 / \log_p q$, откуда сле-

дует практическая формула $m_2 = \left\lceil \frac{m_1}{\log_p q} \right\rceil + 1$. Далее выполняем округление по последнему разряду, после чего этот последний разряд отбрасывается.

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

ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНЫХ СИСТЕМ СЧИСЛЕНИЯ

Использование вспомогательных систем счисления позволяет ускорить процесс перевода чисел. В сферах, связанных с вычислительной техникой, вспомогательные СС имеют основания, кратные степени двойки (четверичная, восьмеричная, шестнадцатеричная)

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

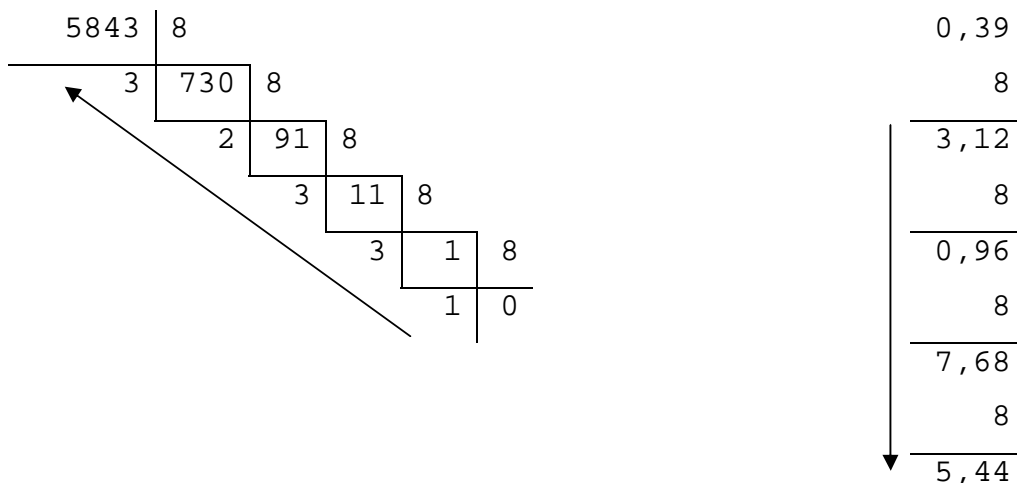
Сформулируем правило перевода чисел из 10СС в 2СС с использованием 8СС (16СС) и обратно с использованием вспомогательных СС.

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

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

Вправо и влево от запятой число в 2СС разбивается на триады (тетрады), а их заменяют соответствующими цифрами 8СС (16СС). 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.



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

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

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

ФОРМАТЫ ДАННЫХ В ЭВМ

Любая информация – числа, команды, АЦ-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), слова могут быть разбиты на части. Со-

временные ЭВМ имеют байториентированную адресацию памяти: 1 байт = 8 бит. Введены понятия «полуслово» (2 байта=16 бит), «слово» (4 байта=32 бита), «двойное слово» (8 байт=64 бита).

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

Для двоичных чисел с ФЗ используются три формата фиксированной длины: полуслово – короткий с ФЗ, слово – длинный с ФЗ, двойное слово – для промежуточных действий, чтобы обеспечить высокую точность вычислений. Двоичные операнды имеют вид целых чисел в дополнительном коде, у которых крайний левый разряд – знаковый.

Двоичные числа с ПЗ изображаются по-разному в ЕС ЭВМ и СМ ЭВМ. Общим в изображении является лишь то, что порядки имеют смещения. В ЕС ЭВМ для чисел с ПЗ имеются три формата: короткий – слово, длинный – двойное слово и расширенный – учетверенное слово. Во всех этих форматах 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 шестнадцатеричном числе. Мантисса изображается в прямом коде и должна быть нормализована.

В СМЭВМ и ПЭВМ для чисел с ПЗ используются два формата: короткий и длинный. Смещенный порядок занимает восемь разрядов (смещение=128), крайний левый разряд сетки отводится под знак числа, остальные под мантиссу, изображенную в 2СС (23 разряда в коротком и 55 разрядов в длинном формате). Смещенный порядок содержит информацию о положении запятой в двоичной мантиссе числа. Для повышения точности представления мантиссы старший разряд ее, который в нормализованном виде всегда равен «1», может не заноситься в разрядную сетку, а просто подразумеваться.

Сравнение представления мантисс с двоичным и шестнадцатеричным основание показывает существенное расширение диапазона представления чисел в ЕСЭВМ.

Пример 2. Отрицательное число $A = -5843,39$ представить в двух форматах – длинном и коротком, в форме с ФЗ, в коротком формате в форме с ПЗ в разрядной сетке ЕСЭВМ и СМЭВМ.

В предыдущем примере получено двоичное изображение числа

Модифицированные обратный и дополнительный коды (МОК и МДК) имеют для изображения знака два соседних разряда: «+» – «00», «-» – «11». Эти коды используются для обнаружения ситуации 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.

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

Для исправления результата можно либо прекратить вычисления и выдать на пульт управления машин сигнал ПРС, либо пересмотреть масштаб результата, сдвинув его вправо на один разряд, а в освободившийся старший знаковый разряд поместить значение 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. Числа с ПЗ изображаются двумя частями – мантиссой и порядком:

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

Чтобы сложить их, надо выполнить различные действия над мантиссами и порядками. Поэтому в машинах предусмотрены различные устройства для обработки мантисс и порядков. Мантиссы исходных операндов нормализованы.

1. Выравнивание порядков слагаемых: меньший порядок увеличивает 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 результата, если необходимо. При этом **денормализация вправо**, т.е. ситуация, когда в старшем разряде двоичной мантиссы «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 «1» порядка суммы.

При больших величинах порядков возможно возникновении истинного ПРС числа с ПЗ, хотя вероятность этого невелика.

Смещенные порядки используются в большинстве современных ЭВМ для упрощения процесса выравнивания порядков и их сравнения.

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, и так как они все оказываются целыми положительными числами, то алгебраическое сложение можно производить без предварительного анализа знаков.

Например, изобразим в ЕСЭВМ характеристику, соответствующую порядку (-61):

$$61_{(10)} = 111101_{(2)}: \text{порядок} = (-61) = \underline{1111101}$$

$$(-61)_{\text{дк}} = \underline{1000011}.$$

$$\text{Характеристика} = 64 - 61 = 0000011 = +3_{(10)}.$$

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. Масштаб $M=2^7$.

$$A_{\text{ПК}} = 1,0011110$$

$$B_{\text{ПК}} = 1,1001000$$

$$(A+B)_{\text{ОК}} = 1,0011001$$

$$(A+B)_{\text{ПК}} = 1,1100110 \quad (M=2^7)$$

$$A+B = -1100110 = -102$$

$$A_{\text{ОК}} = 1,1100001$$

$$B_{\text{ОК}} = 1,0110111$$

$$\begin{array}{r} 11,0011000 \\ \hline \end{array}$$

$$\begin{array}{r} 1 \\ \searrow \end{array}$$

$$(A+B)_{\text{ОК}} = 1,0011001$$

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

– Выравниваем порядки, для чего выполняем их вычитание с использованием ДК:

$$P_a = 0101$$

$$P_b = 1001$$

$$(P_a - P_b)_{\text{ДК}} = 1010.$$

Разность – 2, следовательно, мантиссу числа A надо сдвинуть на два разряда вправо и соответственно увеличить порядок:

Мантисса	Порядок
A=1,0011110	0111

Сложить мантиссы в ДК

$$1,1100010 = m_a$$

$$0,1001000 = m_b$$

$$\pm 0,0101010 = m_b + m_a$$

– Нормализовать мантиссу результата, сдвинув ее на один разряд влево и вычтя единицу из порядка.

Мантисса	Порядок
A=0,1010100	0110

$$\text{Проверка: } 0,1010100 \cdot 2^6 = 42.$$

УМНОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ

Процесс умножения чисел в двоичной системе счисления прост, так как разрядами множителя могут быть либо «0», либо «1», и, 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 накопленной суммой частичных произведений, если цифра множителя «1»;
- сдвиги в каждом такте умножения.

Умножение можно выполнять как с младших, так и со старших разрядов множителя, со сдвигом как частичной суммы, так и множимого в процессе умножения. Этим объясняется существование четырех способов 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 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 обычно сдвигается в ту же сторону, что и множитель, а множимое сдвигается навстречу множителю, т.е. в противоположную сторону.

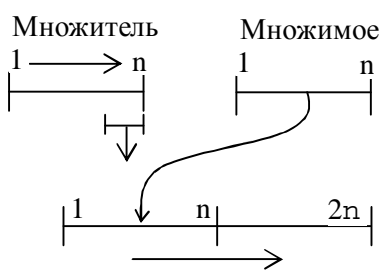
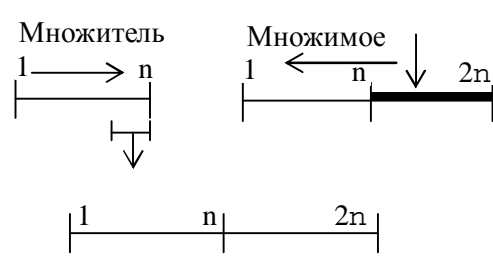
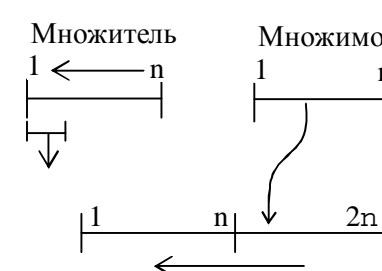
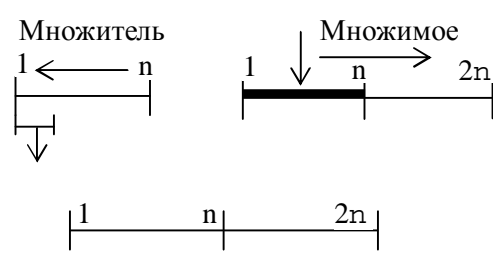
	Сдвиг суммы частичных произведений	Сдвиг множимого
Сдвиг множителя вправо	I способ	II способ
	 <p>Множитель</p> <p>Множимое</p> <p>Сумма частичных произведений</p>	 <p>Множитель</p> <p>Множимое</p> <p>Сумма частичных произведений</p>
Сдвиг множителя влево	III способ	IV способ
	 <p>Множитель</p> <p>Множимое</p> <p>Сумма частичных произведений</p>	 <p>Множитель</p> <p>Множимое</p> <p>Сумма частичных произведений</p>

Рис.1. Схемы четырех способов умножения чисел

I способ – умножение с младших разрядов множителя со сдвигом
суммы частичных произведений вправо

Устройства, которые хранят операнды, регистры, имеют следующую разрядность:

- регистры множителя и множимого – n -разрядные;
- регистр частичных произведений – $2n$ -разрядный.

На схеме показано, что суммирование множимого следует выполнять в старшие n разрядах регистра суммы частичных произведений. Причем разрядность его можно уменьшить вдвое, до n -разрядов, помещая при сдвиге младшие разряды суммы на место освобождающихся разрядов регистра множителя.

Особенность I способа умножения состоит в том, что имеется возможность **временного переполнения 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 два n -разрядных регистра множителя и множимого и одного $2n$ -разрядных регистра суммы частичных произведений. На схеме видно, что суммирование множимого следует выполнять в младшие n разрядов регистра суммы частичных произведений.

Особенность III способа умножения состоит в том, что в последнем такте **не следует выполнять сдвиг** в регистре сумм 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 произведе-

ний. Причем первоначально множимое помещается в старшие разряды регистра, а затем в каждом такте сдвигается на один разряд вправо.

Особенность IV способа умножения состоит в том, что перед началом цикла умножения следует **множимое сдвинуть** на один разряд вправо.

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

Рассмотрим эти алгоритмы для 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. Найти произведение операндов

$A=26_{(10)}=11010_{(2)}$ и $B=-19_{(10)}=-10011_{(2)}$, представленных в 2СС, ПК, с ФЗ.

$M=2^5$	$A=0,11010$
	$B=1,10011$

1. Знак произведения: $0 \oplus 1 = 1$.
2. Перемножим модули сомножителей, используя I способ.
 $V=0,10011$ – модуль множимого.

Таблица

 Множитель	 Сумма ЧП	Пояснения
0,1101 <u>0</u>	0,0000000000	Сдвиги
0,0110 <u>1</u>	0,0000000000	
	0,10011	
	0,1001100000	Сложение
0,0011 <u>0</u>	0,0100110000	Сдвиги
0,0001 <u>1</u>	0,0010011000	Сдвиги
	0,10011	
	0,1011111000	Сложение
	0,0101111100	Сдвиги
0,0000 <u>1</u>	0,10011	
	0,1111011100	Сложение
0,00000	0,0111101110	Сдвиги

3. Прямой код произведения:

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

4. Проверка результата.

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

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

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 псевдопроизведению 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. Перемножим сомножители $A = -18_{(10)} = -10010_{(2)}$ и $B = 27_{(10)} = 11011_{(2)}$, представив их в ДК, используя алгоритм с простой коррекцией.

$$M=2^5 \quad \begin{array}{ll} A_{ПК} = 1,10010; & A_{ДК} = 1,01110 \\ B_{ПК} = 0,11011; & B_{ДК} = 0,11011 \end{array}$$

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

2. Перемножим модули сомножителей, используя II способ.

Таблица

→ Множитель	← Множимое	Сумма ЧП	Пояснения
0,0111 <u>0</u>	0,0000011011	0,0000000000	Сдвиги
0,0011 <u>1</u>	0,0000110110	0,0000110110	
0,0001 <u>1</u>	0,0001101100	0,0000110110 0,0001101100	Сложение Сдвиги
0,0000 <u>1</u>	0,0011011000	0,0010100010	Сложение
0,0000 <u>0</u>	0,011011000	0,0011011000	Сдвиги
0,00000	0,1101100000	0,0101111010	Сложение Сдвиги Сдвиги

Получено псевдопроизведение: 0,0101111010

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

$$\begin{array}{r} B_{ДК} = 0,00101 \qquad \qquad \qquad 0,0101111010 \\ \qquad \qquad \qquad \qquad \qquad \qquad 0,00101 \\ \hline \text{Модуль полного произведения} \quad 0,1000011010. \end{array}$$

4. Полное произведение в дополнительном коде:

$$(A \cdot B)_{ДК} = 1,1000011010.$$

5. Проверка результата.

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

$(A \cdot B)_{\text{ПК}} = 1,0111100110 \quad (\times 2^{10})$.

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

**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 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 **ну-**

лями, при сдвиге **вправо** освобождающиеся старшие разряды заполняются **единицами**, т.е. реализуется арифметический сдвиг числа.

Примерб. Перемножим сомножители $A = -18_{(10)} = -10010_{(2)}$; $B = 27_{(10)} = 11011_{(2)}$, представив их в ДК, используя алгоритм с автоматической коррекцией и I и IV способы умножения.

$$M = 2^5 \quad \begin{array}{ll} A_{\text{ПК}} = 1,10010; & A_{\text{ДК}} = 1,01110 \quad - \text{множимое.} \\ B_{\text{ПК}} = 0,11011; & B_{\text{ДК}} = 0,11011 \quad - \text{множитель.} \end{array}$$

Таблица

IV способ

← Множитель	→ Множимое	Сумма ЧП	Пояснения
<u>0,1</u> 1011 <u>1,1</u> 0110 <u>1,0</u> 1100	1,0111000000 1,1011100000 1,1101110000	0,0000000000 1,0111000000 1,0111000000	Сложение Сдвиги Сдвиги
		1,0111000000	
		1,0111000000	
		0,0010010000	
<u>0,1</u> 1000 <u>1,1</u> 0000 <u>1,0</u> 0000	1,1110111000 1,1111011100 1,1111101110	1,1001010000 1,1001010000 1,1110111000	Вычитание Сдвиги Сложение Сдвиги
		≠1,1000001000	
		1,1000001000 0,0000010010	
0,00000	1,111111011	1,1000010010	Вычитание Сдвиги

Получено произведение в дополнительном коде:

$$(A \cdot B)_{\text{ДК}} = 1,1000011010.$$

Проверка результата.

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

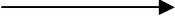
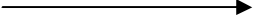
$$(A \cdot B)_{\text{ПК}} = 1,0111100110 \quad (\times 2^{10}).$$

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

I способ умножения позволяет обратить внимание на необходимость сохранения предыдущей цифры множителя при сдвиге его вправо (в первом такте соседней младшей цифрой всегда является «0»).

Таблица

I способ

 Множитель	 Сумма ЧП	Пояснения
0,1101 <u>10</u>	0,0000000000 0,10010	Вычитание Сдвиги Сдвиги Сложение Сдвиги
0,0110 <u>11</u>	0,1001000000 0,0100100000	
0,0011 <u>01</u>	0,0010010000 1,01110	
0,0001 <u>10</u>	1,1001010000 1,1100101000 0,10010	
0,0000 <u>11</u>	0 ,0101101000 0,0010110100	
0,0000 <u>01</u>	0,0001011010 1,01110 1,1000011010	
		Нет последнего сдвига!

Получено произведение в дополнительном коде:

$$(A \cdot B)_{\text{ДК}} = 1,1000011010.$$

Проверка результата.

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

$$(A \cdot B)_{\text{ПК}} = 1,0111100110 \quad (\times 2^{10}).$$

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

УМНОЖЕНИЕ ЧИСЕЛ В ФОРМЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Когда сомножители заданы в форме с ПЗ

$$A = \pm m_A \cdot 2^{\pm P_A}, \quad B = \pm m_B \cdot 2^{\pm P_B},$$

то их произведение определяется следующим образом:

$$C = A \times B = \pm m_A \cdot m_B \cdot 2^{\pm(P_A + P_B)} = \pm m_C \cdot 2^{\pm P_C},$$

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 модифицированного дополнительного или обратного кодов.

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 возникновение ПРС при сложении порядков, поэтому необходимо предусматривать выявление признаков ПРС в устройствах умножения чисел с ПЗ.

Пример 7. Операнды $A=26_{(10)}$ и $B=-19_{(10)}$, представить в форме с ПЗ в разрядной сетке условной машины и перемножить, используя при умножении мантисс III способ.

A = 11010₍₂₎. Операнды в разрядной
 B = -10011₍₂₎. сетке условной машины

0	11010	0	0101
1	10011	0	0101
знак числа	Мантисса пять разрядов	знак порядка	Порядок четыре раз- ряда

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

2. Произведение модулей мантисс получим, используя III способ умножения и считая B множимым.

Таблица

← Множитель	← Сумма ЧП	Пояснения
0, <u>1</u> 1010	0,0000000000 10011	Сложение
0, <u>1</u> 0100	0,0000010011 0,0000100110 10011	
0, <u>0</u> 1000	0,0000111001 0,0001110010 0,0011100100 10011	Сложение Сдвиги Сдвиги
0, <u>1</u> 0000	0,0011110111	Сложение
0, <u>0</u> 0000	0,0111101110	Сдвиги Нет последнего сдвига!

Полное 2n-разрядное произведение модулей мантисс: 0,0111101110.

3. Порядок произведения

0	0101
0	0101
<hr/>	
0	1010.

Так как в разрядной сетке условной машины под мантиссу отведено n разрядов, то необходимо округление мантиссы результата, что приводит к погрешности.

Результат в разрядной сетке

1	01111	0	1010
---	-------	---	------

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

1	11110	0	1001
---	-------	---	------

Проверка: $-0,11110 \cdot 2^9 = -111100000_{(2)} = -480_{(10)}$. Ошибка округления равна $-14_{(10)}$, именно это число и было получено в младших n разрядах полного произведения мантисс, которые были отброшены при округлении.

Примечание. Для уменьшения погрешности округления следует сначала выполнять нормализацию, а затем округление, тогда больше верных значащих цифр мантиссы произведения попадет в разрядную сетку машины.

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.

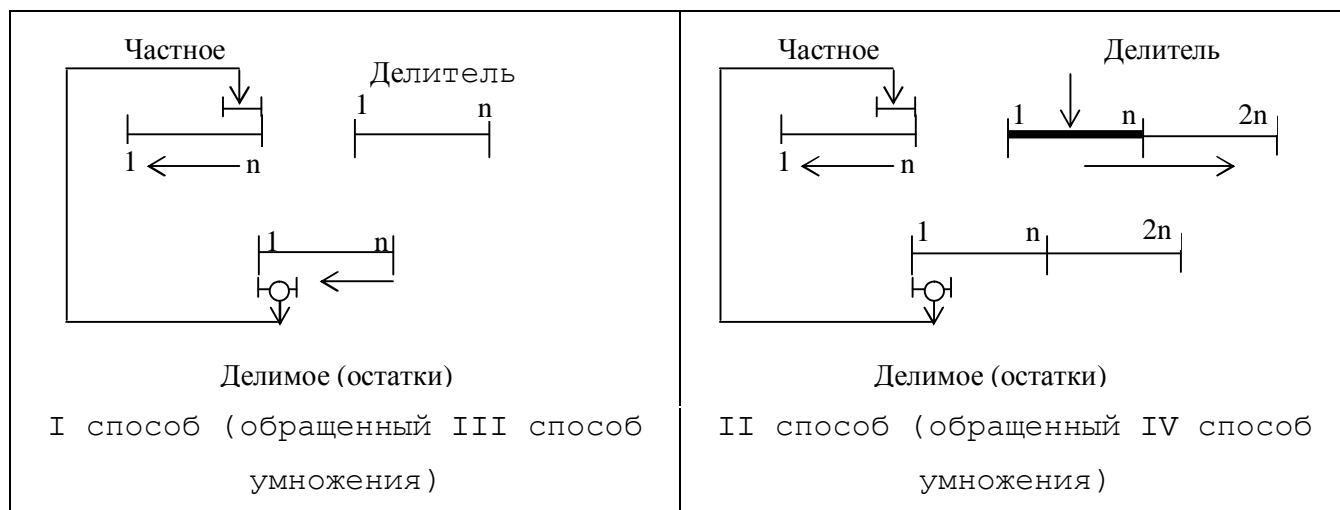


Рис. 2. - Схемы способов деления чисел

ОСОБЕННОСТИ ДЕЛЕНИЯ ДРОБНЫХ И ЦЕЛЫХ ЧИСЕЛ.

Известно, что двоичные числа с ФЗ можно промасштабировать и рассматривать их как дробные или как целые.

При делении *дробных чисел*: если делимое больше делителя, в результате должно быть получено число большее единицы, что является признаком ПРС, следовательно, операцию деления необходимо прекратить. Признаком ПРС является положительный или равный нулю остаток после самого первого вычитания.

При делении *целых чисел* ситуация с ПРС не возникает, однако, возможно получить частное, равное нулю. Для ускорения деления целых чисел (в общем случае старшие разряды делимого и делителя могут оказаться незначущими) выполняют синхронные сдвиги влево в регистрах делимого и делителя. И если первая значащая цифра появится в регистре делимого, процесс деления выполняют далее в соответствии с выбранным способом и алгоритмом. Если же первая значащая цифра появится в регистре делителя, значит делитель больше делимого и целого числа в процессе деления получить нельзя, следовательно, частное равно нулю, и операция деления не выполняется.

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 заносится «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 вправо (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 п.4.

9. Завершить цикл формированием $(n+1)$ -го разряда остатка для округления частного.

10. Выполнить округление результата и присвоить частному знак согласно п.1.

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 – $(\Delta_i + D)$ или невосстановленного Δ_i):

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

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

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

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. Разделить $A=27_{(10)}$ на $B=-30_{(10)}$, операнды в форме с ФЗ в прямом коде $M_{A,B}=2^5$.

$A=0,11101$ – делимое; $B=1,11110$ – делитель.

1. Знак частного: $0 \oplus 1 = 1$.

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

Таблица

← Частное	← Делимое (остатки)	Пояснения
0,00000	0,11011 1,00010	Вычитание
0,0000 <u>0</u> ←	1,11101 0,11110	Первый остаток Восстановление
0,000 <u>01</u>	0,11011 1,10110 1,00010	Сдвиги Вычитание
0,00 <u>011</u>	0,11000 1,10000 1,000010	Второй остаток Сдвиги Вычитание
0,0 <u>0111</u> ←	0,10010 1,00100 1,00010	Третий остаток Сдвиги Вычитание
0, <u>01110</u> ←	0,00110 0,01100 1,00010 1,01110 0,11110	Четвертый остаток Сдвиги Вычитание Пятый остаток Восстановление
0, <u>11100</u> ←	0,01100 0,11000 1,00010 1,11010 0,11110 0,11000 1,10000 1,00010 0,100010	Сдвиги Вычитание Шестой остаток Восстановление Сдвиг остатка Вычитание Седьмой остаток для округления

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

Тогда модуль частного после округления:

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

4. Частное со знаком в прямом коде: 1,11101.

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

Точный результат $-(27/30) = -0,9$.

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

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

$A = -1,11001$ – делимое; $B = 0,01001$ – делитель.

1. Знак частного: $1 \oplus 0 = 1$.

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

Таблица

← Частное	→ Делитель	Делимое (остатки)	Пояснения
	0,0100100000	0,1100100000 1,1011100000	
		≠0,1000000000	ПРС!
Увеличим масштаб делимого до $M=2^7$			
0,0000 <u>0</u> ←	0,0100100000	0,00110 01000 1,10111 00000	Вычитание Первый остаток
		1,11101 01000	
0,000 <u>01</u> ←	0,0010010000	0,01001 00000	Восстановление Сдвиги Вычитание
		≠0,00110 01000	
		0,00110 01000 1,11011 10000	
0,00 <u>010</u> ←	0,0001001000	≠0,00001 11000	Второй остаток Сдвиги
		0,00001 11000 1,11101 11000	
0,00 <u>010</u> ←	0,0000100100	1,11111 10000	Третий остаток Восстановление
		0,00010 01000	
		≠0,00001 11000	
		0,00001 11000 1,11110 11100	Сдвиги Вычитание

Продолжение таблицы

<div>←</div> <div>Частное</div>	<div>→</div> <div>Делитель</div>	Делимое (остатки)	Пояснения	
0,0 <u>0101</u> ←	0,0000010010	<div>≠0,00000 10100</div> <div>0,00000 10100</div> <div>1,11111 01110</div>	Четвертый остаток Сдвиги Вычитание	
0,0 <u>01011</u> ←		0,0000001001	<div>≠0,00000 00010</div> <div>0,00000 00010</div> <div>1,11111 10111</div>	Пятый остаток Сдвиги Вычитание
0, <u>10110</u> ←			0,0000000100	<div>1,11111 11001</div> <div>0,00000 01001</div>
	<div>0,00000 00010</div> <div>0,00000 00010</div> <div>1,11111 11100</div>	Сдвиги Вычитание		
	0, <u>10110</u> (0) ←	<div>1,11111 11110</div>		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 $-(25/9) = -2,78$.

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).

Для I способа деления после упрощения второй строки в формуле (1) получим

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

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.

Для II способа деления после упрощения второй строки в формуле (2) получим

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

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 пп. 4-6 алгоритма (n+1) раз, причем, последний сдвиг частного не выполнять, так как (n+1)-й разряд формируется для округления.

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

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

$A=1,11101$ – делимое; $B=1,11110$ – делитель.

1. Знак частного: $1 \oplus 1 = 0$.

2. Деление модулей операндов выполняется I способом.

Таблица

← Частное	← Делимое (остатки)	Пояснения
0,00000	0,01100 1,01101	Вычитание
0,0000 <u>0</u> ←	1,11001 1,10011 0,10010 10,00101 1	Первый остаток < 0 Сдвиги Сложение
0,000 <u>01</u> ←	0,00110 0,01100 1,01101	Второй остаток > 0 Сдвиги Вычитание
0,00 <u>010</u> ←	1,11001 1,10011 0,10010 10,00101 1	Третий остаток < 0 Сдвиги Сложение
0,0 <u>0101</u> ←	0,00110 0,01100 1,01101	Четвертый остаток > 0 Сдвиги Вычитание
0, <u>01010</u> ←	1,11001 1,10011 0,10010	Пятый остаток < 0 Сдвиги Сложение
	10,00101 1	

← Частное	← Делимое (остатки)	Пояснения
0, <u>10101</u> ← 0, <u>10101</u> (0)	0,00110	Шестой остаток > 0
	0,01100	Сдвиги
	1,01101	Вычитание
	1,11001	Седьмой остаток

3. Модуль частного после округления:

$$A/B = 0,10101.$$

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

Точный результат $(12/18) = 0,667.$

Замечание. Следует обратить внимание на особенности использования ОК при вычитании: при сдвиге отрицательных чисел как влево, так и вправо свободные разряды заполняются «1». Кроме того, в соответствии с правилами сложения в ОК возникающую единицу переноса из знакового разряда 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. Ранее были рассмотрены алгоритмы умножения в ДК, здесь познакомимся с алгоритмом деления 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 к 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 частного 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 пп. 4-6 алгоритма (n+1) раз, с учетом формирования разряда частного для округления (последний частного сдвиг не выполнять!).

8. Выполнить округление результата.

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

$A = 1,11010$ - делимое; $B = 0,11101$ - делитель.

Переводим в дополнительный код:

$A = 1,00110$ - делимое; $B = 0,11101$ - делитель.

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

Таблица

← Частное	← Делимое (остатки)	Пояснения
0,0000 <u>1</u>	1,00110 0,11101	Сравнение знаков Сложение
← Частное	← Делимое (остатки)	Пояснения
0,000 <u>10</u>	10,00011 0,00110 1,00011	1-й ост. - нет ПРС Сдвиги, сравнение знаков В и 1-й ост. Вычитание
0,00 <u>100</u>	1,01001 0,10010 0,11101	2-ой ост., в частное - 0 Сдвиги, сравнение знаков В и 2-й ост. Сложение
0,0 <u>1000</u>	1,01111 0,11110 0,11101	3-й остаток, в частное - 0 Сдвиги, сравнение знаков В и 3-й ост. Сложение

← Частное	← Делимое (остатки)	Пояснения
<div>0, <u>10001</u></div> <div><u>1,00011</u></div> <div><u>1,00011</u>(0)</div>	<div>1,11011</div> <div>1,10110</div> <div>0,11101</div>	<div>4-й остаток, в частное – 0</div> <div>Сдвиги, сравнение знаков В и 4-й ост.</div> <div>Сложение</div>
	<div>10,10011</div> <div>1,00110</div> <div>1,00011</div>	<div>5-й остаток, в частное – «1»</div> <div>Сдвиги, сравнение знаков В и 5-й ост.</div> <div>Вычитание</div>
	<div>10,01001</div> <div>0,10010</div> <div>1,00011</div>	<div>6-ой остаток, в частное – «1»</div> <div>Сдвиг остатка</div> <div>Вычитание</div>
	<div>1,10101</div>	<div>7-й остаток, в частное – «1»</div>

3. Модуль частного после округления:

$$A/V=1,00011.$$

Проверка: частное в прямом коде $A/V=1,11101$;

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

Точный результат $-(26/29)=0,89655$.

ДЕЛЕНИЕ ЧИСЕЛ В ФОРМЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Когда операнды заданы в форме с ПЗ

$$A = \pm m_A \cdot 2^{\pm P_A}, \quad B = \pm m_B \cdot 2^{\pm P_B},$$

то их частное определяется следующим образом:

$$C = A \times B = \pm m_A / m_B \cdot 2^{\pm(P_A - P_B)} = \pm m_C \cdot 2^{\pm P_C},$$

т.е. мантисса частного m_C есть частное мантисс делимого и делителя, а порядок P_C – 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 п.1 данного алгоритма.

В отличие от деления чисел в форме с ФЗ при выполнении п.2 данного алгоритма получение положительного остатка при первом вычитании не означает ПРС. При этом достаточно денормализовать мантиссу делимого сдвигом ее на один разряд вправо с одновременным увеличением на один порядка делимого.

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

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

$A = 11010_{(2)}$ Операнды в разрядной
 $B = -10011_{(2)}$ сетке условной машины

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

1. Знак частного: $0 \oplus 1 = 1$.

2. Частное от деления модулей мантисс получим, используя II способ.

Таблица

<div>←</div> <div>Частное</div>	<div>→</div> <div>Делитель</div>	Делимое (остатки)				Пояснения		
	0,1001100000	0,11010 00000				1-е вычитание		
		1,01100 11111						
		0,00110 11111 <div>1</div>				1-й ост.>0 – денормализация		
0,00111 00000								
Денормализация мантиссы делимого		0	01101	0	110	А		
<div>0,0000<u>0</u></div> <div>↓</div> <div>0,000<u>01</u></div> <div>↓</div> <div>0,00<u>010</u></div> <div>↓</div> <div>0,0<u>0101</u></div>	0,1001100000	0,01101 00000				Вычитание		
		1,01100 11111						
	0,0100110000	1,11001 11111				1-й ост.<0		
		1,11001 11111						
		0,01001 10000				Сдвиги		
		10,00010 01111						
			1				Сложение	
	0,00010 10000	0,00010 10000				2-й ост.>0		
		0,0010011000	0,00010 10000					
			1,11011 00111				Сдвиги	
	0,0001001100	1,11101 10111				Вычитание		
		1,11101 10111						
		0,00010 01100				3-й ост.<0		
	10,00000 00011				Сдвиги			
	1							
0,0000001001	0,00000 00100				4-й ост.>0			
	0,00000 00100							
		1,1111011001				Сдвиги		
0,0000001001	1,11111 10000				Вычитание			
	1,11111 10000							
		0,00000 01001				6-й ост.<0		
0,0000001001	1,11111 11001				Сдвиги			
	0,00000 01001							
		1,11111 11001				Сложение		
0,0000001001	1,11111 11001				7-й ост.<0			
	0,00000 01001							
		1,11111 11001				Сложение		
0,0000001001	1,11111 11001				7-й ост.<0			
	0,00000 01001							

3. Вычитание порядков в ДК:

00110

11011

~~1~~00001 -порядок частного.

4. Мантисса нормализована. Результат в разрядной сетке:

1	10100	0	0001
---	-------	---	------

Проверка: $-0,10100 \cdot 2^1 = -1,01_{(2)} = -1,25_{(10)}$.

Точный результат $A/B = -(26/19) = -1,36_{(10)}$.

КАЛЕНДАРНЫЙ ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Содержание этапа	Объем %	Срок завершения (неделя)
Сложение двоичных чисел	10%	1-я неделя
Умножение двоичных чисел	20%	4-я неделя
Деление двоичных чисел	20%	7-я неделя
Сложение двоично-десятичных чисел	15%	10-я неделя
Умножение двоично-десятичных чисел	20%	12-я неделя
Оформление ПЗ	15%	15-я неделя

Защита курсовой работы не позднее 16-17-й недели.

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 – Киров:Изд-во ВятГТУ 2001.- 36 с.