Выполнение арифметических операций:

- сложение;
- вычитание;
- умножение

для двоичных чисел в форме :

- с фиксированной запятой;
- с плавающей запятой.

Сложение двоичных чисел с фиксированной запятой

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

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

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

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

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

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

Переполнение разрядной сетки

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

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

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

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

Суммируем числа в дополнительном коде:

$$
\begin{array}{r}
1\ 01001 \\
+\ \underline{1\ 01110} \\
0\ 10111
\end{array}
$$

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

Способы выявления переполнения в арифметических операциях:

1) Анализируются два переноса – из старшего значащего разряда в знаковый (р1) и из знакового разряда (р2).

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

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

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

Способы выявления переполнения в арифметических операциях:

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

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

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

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

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

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

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

Способы выявления переполнения в арифметических операциях:

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

$A_{1\ доп}$ = 0 01001 (+23);

$A_{2\ доп}$ = 0 01110 (+18).

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

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

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

Способы выявления переполнения в арифметических операциях:

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

ЭВМ вырабатывает сигнал φ = 1 (Останов).

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

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

$$\varphi = \begin{cases} 1, \textit{если } \textit{зн}1 \neq \textit{зн}2, \\ 0, \textit{если } \textit{зн}1 = \textit{зн}2. \end{cases}$$

Алгоритм сложения чисел с фиксированной запятой

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

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

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

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

Пример сложения чисел с фиксированной запятой

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

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

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

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

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

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

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

Проверка: $(-6) + (-9) = (-15)$.

Вычитание двоичных чисел с фиксированной запятой

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

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

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

Представление двоичных чисел с плавающей запятой

Число А в форме с плавающей запятой представляется в виде

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

где m_n – нормализованная мантисса числа А;
Р – порядок (характеристика) числа А;
q – основание системы счисления.

Мантисса m_n представляет собой правильную дробь, удовлетворяющую условию

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

Числа А1 и А2 представлены следующим образом:

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

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

Алгоритм сложения двух чисел с плавающей запятой:

1. Выравнивание порядков суммируемых чисел:

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

$P_1 – P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Нормализация мантиссы

1. Выравнивание порядков суммируемых чисел:

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

$P_1 - P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Нарушение нормализации мантиссы вправо

1. Выравнивание порядков суммируемых чисел:

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

$P_1 – P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Нарушение нормализации мантиссы влево

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

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

Примеры для доп. и обр. кодов:

$M_x = 01{,}110111$, $P_x = 00{,}101$

После нормализации:

$M_x = 00{,}111011$ ①*, $P_x = 00{,}110$

$M_x = 10{,}10011$, $P_x = 00{,}011$

После нормализации:

$M_x = 11{,}01001$ ①*, $P_x = 00{,}100$

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

Умножение двоичных чисел в форме с фиксированной запятой

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

Умножение, начиная с младшего разряда множителя:

а) дробные числа

```
*  X =      0,011
   Y =      0,101
   ------------------
             011  ← Слагаемое 1
      +     000   ← Слагаемое 2
           011    ← Слагаемое 3
   ------------------
   Z =  0,001111
```

б) целые числа

```
*  X =        011
   Y =        101
   ------------------
              011  ← Слагаемое 1
      +      000   ← Слагаемое 2
            011    ← Слагаемое 3
   ------------------
   Z =      01111
```

Умножение двоичных чисел в форме с фиксированной запятой

Умножение, начиная со старшего разряда множителя:

а) дробные числа

```
*  X =    0,011
   Y =    0,101
   ------------
           011      ← Слагаемое 1
      +     000     ← Слагаемое 2
             011    ← Слагаемое 3
          ------
   Z =  0,001111
```

б) целые числа

```
*  X =      011
   Y =      101
   ------------
            011     ← Слагаемое 1
      +      000    ← Слагаемое 2
              011   ← Слагаемое 3
           ------
   Z =      01111
```

Умножение двоичных чисел в форме с фиксированной запятой

1. Выравнивание порядков суммируемых чисел:

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

$P_1 – P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Умножение двоичных чисел в форме с фиксированной запятой

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

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

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

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

Умножение двоичных чисел в прямом коде

1. Выравнивание порядков суммируемых чисел:

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

$P_1 - P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Алгоритм умножения операндов в прямых кодах

1. Выравнивание порядков суммируемых чисел:

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

$P_1 – P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Пример умножения операндов в прямых кодах

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

$A_{пр}$ = 10110 = $(–6)_{10}$;

$B_{пр}$ = 11101 = $(–13)_{10}$.

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

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

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

Пример умножения операндов в прямых кодах

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

0000000000 – $\Sigma ЧП_0$;

+ 00110 – прибавление множимого, разряд множителя равен 1;

0011000000 – $\Sigma ЧП_1$;

0001100000 – сдвиг вправо на 1 разряд $\Sigma ЧП_1$;

0000110000 – сдвиг вправо на 1 разряд $\Sigma ЧП_2$;

+ 00110 – прибавление множимого, разряд множителя равен 1;

0011110000 – $\Sigma ЧП_3$;

0001111000 – сдвиг вправо на 1 разряд $\Sigma ЧП_3$;

+ 00110 – прибавление множимого, разряд множителя равен 1;

0100111000 – $\Sigma ЧП_4$;

0010011100 – сдвиг вправо на 1 разряд $\Sigma ЧП_4$;

0001001110 – дополнительный сдвиг вправо на 1 разряд после умножения на все значащие разряды множителя для правильной постановки результата в формате 2n разрядов (или умножение на знаковый разряд).

Алгоритм умножения операндов в прямых кодах

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

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

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

Алгоритм умножения операндов в дополнительных кодах

1. Выравнивание порядков суммируемых чисел:

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

$P_1 – P_2 = \Delta P$;

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

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

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

3. Результату присваивается больший порядок.

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

Алгоритм умножения операндов в дополнительных кодах

Следует обратить внимание:

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

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

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

Пример умножения операндов в дополнительных кодах

Пример 2. Умножение в дополнительном коде $Z_{доп} = A_{доп} * B_{доп}$.
$A_{пр} = (+3)_{10} = 0\ 011_{пр} = 0\ 011_{доп} = 00\ 011^{м}_{доп}$;
$B_{пр} = (–5)_{10} = 1\ 101_{пр} = 1\ 011_{доп} = 11\ 011^{м}_{доп}$.

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

$[A]^{м}_{пр} \rightarrow [-A]^{м}_{пр} \rightarrow [-A]^{м}_{доп}$. Таким образом, получается:
$00\ 011 \rightarrow 11\ 011 \rightarrow 11\ 101$. Отсюда $K = [-A]^{м}_{доп} = 11\ 101$.

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

Пример умножения операндов в дополнительных кодах

$*$ 00 011

11 011

00 000000 – $\Sigma ЧП_0$;

+ 00 011 – прибавление множимого, разряд множителя равен 1;

00 011000 – $\Sigma ЧП_1$;

00 001100 – сдвиг вправо на 1 разряд $\Sigma ЧП_1$;

+ 00 011 – прибавление множимого, разряд множителя равен 1;

00 100100 – $\Sigma ЧП_2$;

00 010010 – сдвиг вправо на 1 разряд $\Sigma ЧП_2$;

00 001001 – сдвиг вправо на 1 разряд $\Sigma ЧП_3$;

+ 11 101 – корректирующая поправка $K = [-A]^{м}_{доп}$;

11 110001 – результат умножения $Z^{м}_{доп}$.

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

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

Умножение и деление операндов с плавающей запятой

Числа с плавающей запятой A1 и A2 представлены следующим образом:

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

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

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

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

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