Лабораторная работа №4

**Моделирование работы операционного автомата, выполняющего операцию сложение двоичных чисел с плавающей запятой.**

**Цель работы:** изучить структуру операционного автомата, выполняющего операцию сложение двоичных чисел с плавающей запятой, на основе данной структуры создать модель операционного автомата, выполняющего операцию суммирования в дополнительном коде с плавающей запятой.

**Задача:** По представленной блок-схеме алгоритма: сложение чисел с плавающей запятой в дополнительном коде, протестировать программу на языке высокого уровня.

1. Теоретическая часть.

**1.1 Представление чисел с плавающей запятой.** Оно основывается на изображении чисел в полулогарифмической форме ,соответствующей записи чисел в нормальной форме, , где p-целое число, называемое порядком числа А; d-основание системы счисления; М-мантисса числа А (обычно |M| < 1).

Фактически положение запятой в мантиссе М определяется величиной порядка р. С изменением р в большую или меньшую сторону запятая соответственно перемещается влево или вправо, т.е. «плавает» в изображении числа.

Пример 1. Представить в форме с плавающей запятой числа  и .

Решение. 1..

2. ..

Числа, удовлетворяющие условию 1/d ≤ M ≤ 1, называют нормализованными.

Как видно из примера, при разных порядках числа положение запятой бывает различным; поэтому ЭВМ, допускающие такую форму записи числа, относят к машинам с плавающей запятой. В их памяти числа хранятся нормализованными. В разрядной сетке ЭВМ (рис.1) фиксируются знак числа, знак порядка, порядок числа и числовое выражение мантисс.



Нормализованное представление чисел позволяет сохранять в разрядной сетке большое количество значащих цифр, что повышает точность вычислений. Обычно в ЭВМ нормализация как при вводе чисел, так и в процессе вычислений (после выполнения очередной операции) осуществляется автоматически. При этом мантисса сдвигается влево на необходимое число разрядов и производится соответствующее уменьшение порядка, т.е. производится «нормализация влево». При выполнении операции сложения или вычитания нормализованных чисел с разными порядками одно из них «денормализуется» до уравнения порядков, а сумма (или разность) опять нормализуется.

В ЭВМ с плавающей запятой возможно переполнение разрядной сетки, также как и в машине с фиксированной запятой. Например, переполнение может возникнуть при сложении нормализованных чисел с одинаковыми порядками. В этом случае появляется единица слева от запятой. Такого рода переполнение исключается сдвигом мантиссы вправо на один разряд и увеличением порядка на единицу, т.е. производится «нормализация вправо».

От количества разрядов, отводимых для хранения порядка, зависит диапазон записываемых чисел, а от количества разрядов, отводимых для мантиссы, - точность записи числа. Если под цифровые разряды мантиссы отведено n, а порядка – m разрядов, то максимальное и минимальное по абсолютной величине нормализованные двоичные числа соответственно равны:

 и  (1)

Диапазон чисел в ЭВМ с плавающей запятой определяется неравенством

 (2)

При достаточно больших n (обычно n≥30) 1- и неравенство принимает вид

  (3)

Из этого неравенства видно, что диапазон чисел зависит в основном от порядка р. Если число превышает верхний предел неравенства (1), то происходит переполнение разрядной сетки и ЭВМ автоматически останавливается.

Если число выйдет за нижний предел неравенства (1), то оно будет соответствовать машинному нулю. Если m взято достаточно большим, то абсолютная величина А будет настолько велика, что при вычислениях для большинства задач все числа не будут выходить из диапазона чисел ЭВМ и не нужно будет прибегать к масштабным коэффициентам. Однако при этом уменьшается точность вычислений, так как увеличиваются ошибки округления и потеря значности при нормализации. В целях устранения такого недостатка в современных ЭВМ предусматривается возможность перераспределения разрядов машинного слова между мантиссой и порядком программным путем.

В последних моделях ЭВМ получило распространение представление чисел с плавающей запятой с недвоичным основанием системы счисления (d=8 и d=16). Использование недвоичного основания несколько уменьшает точность вычислений при заданном числе разрядов мантиссы, но позволяет увеличить диапазон представляемых чисел в ЭВМ и ускорить выполнение некоторых операций, в частности нормализации, за счет того, что сдвиг производится сразу на несколько двоичных разрядов (при d=8 на три разряда, при d=16 на четыре разряда). Вероятность получения ненормализованных чисел в ходе вычислений значительно уменьшается.

В при представлении чисел с плавающей запятой применяется основание d=16 и порядок числа выражается через характеристик*у* -смещенные порядки, представляющие собой сумму действительного порядка р числа с учетом знака и числа 64, т.е. =р+64.

**

Отрицательные и положительные порядки р, увеличившись на 64, оказываются смещенными на числовой оси в положительном направлении и представляются 7-разрядными двоичными числами без знака от 0 до 127 (рис.2). Так как все характеристики  являются положительными, то это упрощает выполнение некоторых действий над ними, например исключение операций над знаками порядков.

В ЭВМ в операциях с плавающей запятой числа могут быть двух форматов фиксированной длины: короткий – в одно слово (рис.3,а) и длинный – в двойное слово (рис.3,б). Форматы различаются длинной мантиссы. Характеристики и мантиссы положительных и отрицательных чисел хранятся в ОП в прямом коде (рис.3,в).





В некоторых мини- и микроЭВМ предусматривается возможность обрабатывать числа с плавающей запятой. Формат слова данных с плавающей запятой для 16-байтовых ЭВМ показан на рис.4,а. В целях увеличения точности представления дробной части числа и диапазона представления чисел используют для представления числа с плавающей запятой два прилегающих друг к другу 16-битовых слова (рис.4,б). Младшая часть мантиссы хранится в следующем, старшем по адресу 16-битовом слове.

 1.2 Структуры операционных блоков для операций над двоичными числами с плавающей запятой.

Арифметические операции с двоичными числами с плавающей запятой требуют выполнения операций как над мантиссами, так и операндами. Поэтому в ЭВМ используют отдельные операционные блоки для обработки мантисс и порядков или производится их последовательная обработка. Выбор способа обработки связан с требованиями к производительности ЭВМ.

Операционные блоки, выполняющие операции с мантиссами, по своему построению аналогичны *АЛУ* с фиксированной запятой. Операционные блоки, выполняющие операции с порядками, по своему построению несколько проще. Связь *АЛУ* с плавающей запятой с *ЦУУ* ЭВМ осуществляются местными блоками управления мантиссами и порядками.

 Рассмотрим принцип построения и работы операционных блоков для обработки порядков при выполнении арифметических операций над двоичными числами с плавающей запятой.

 **1.3** **Сложение и вычитание двоичных чисел с плавающей запятой**

# Число Х, представленное в форме с плавающей запятой, имеет вид

А = М(А)РУ , (4)

где У – порядок числа; М(А)– мантисса; Р – основание системы счисления.

Число Х называется нормализованным, если его мантисса удовлетворяет условию

 1/Р ≤ М(А) < 1 (5)

или при Р=2

 1/2 ≤ М(А) < 1 (6)

старший разряд мантиссы всегда значащий.

###### Диапазон представления порядка числа лежит в пределах

 -(2k - 1) ≤ У ≤ 2k – 1. (7)

Операция сложение-вычитание чисел с плавающей запятой производиться при равенстве порядков. Первое действие при сложении это выравнивание порядков чисел. После суммирования мантисс, возможно, возникнет ситуация, когда будет нарушена нормализация результата как влево на один разряд (левая денормализация), так и вправо на любое число разрядов (правая денормализация). В этом случае результат необходимо нормализовать. Нормализованный результат, возможно, не поместиться в разрядной сетки машины. Поэтому необходимо выполнить операцию округление результата, а затем снова проверить результат на нормализацию. Операция сложение-вычитание чисел с плавающей запятой выполняется в несколько этапов:

1. Выравнивание порядков.
2. Суммирование мантисс.
3. Определение порядка результата.
4. Нормализация результата.
5. Округление результата.
6. Окончательная нормализация результата.

Пусть заданы два числа А и В. Получить сумму чисел в двоичной системе счисления.

Число А будет представлено в виде:

А = М(А) 2У (8)

Число В также будет представлено в виде:

В = М(В) 2Х (9)

Результатом сложения будет число С, также представленная в формате с плавающей запятой.

С = А + В = М(А) 2У + М(В) 2Х (10)

Выполнение операции будет проходить поэтапно:

1. Производится выравнивание порядков двоичных чисел У и Х. Для этого необходимо определить порядок какого числа меньше. Операция сравнения порядков осуществляется на сумматоре, на котором из порядка первого числа У вычитают порядок второго числа Х. По знаку разности определяют больший порядок, а абсолютная величина разности позволяет определить необходимое число сдвигов. При выполнении операции сложения или вычитания применяются дополнительные или обратные коды. Отнимаем от числа У число Х. Все операции выполняются на сумматоре, при этом число Х будет представлено в дополнительном (обратном) коде. Если разность порядков Δ положительная, то необходимо сдвигать вправо мантиссу числа Х. Если разность порядков Δ отрицательная, то сдвигать вправо надо мантиссу числа У. Если Δ > 0, то при каждом сдвиге из разности порядков необходимо вычитать 1, а если Δ < 0 , то прибавлять 1. В результате выполнения этой операции получается, что порядок меньшего (по модулю) числа принимается равным порядку большего числа, а мантисса меньшего числа сдвигается вправо на число разрядов равное разности порядков чисел, обозначим ее как М′ (В) .
2. Производится сложение (вычитание) мантисс; порядку суммы (разности) присваивается порядок большего числа. Сложение-вычитание мантисс может привести к тому, что результат окажется денормализованным. Признаком левой денормализации при использовании модифицированных кодов служит комбинация чисел 01 и 10 в знаковых разрядах, это означает переполнение разрядной сетки. Нарушением правой нормализации является наличие нулей после запятой в прямом коде – 00,0001 или 11,0000101.

В результате выполнения этих этапов получаем результат:

С = А + В = М(А) 2У + М(В) 2Х = М(А) 2Робщ. + М′(В) 2Робщ. (11)

выносим за скобки общий порядок чисел - 2Робщ.  Получаем:

С = 2Робщ. (М(А) + М′(В) ) = 2Робщ. М(С) (12)

1. Полученная сумма (разность) нормализуется. Нормализация результата осуществляется путем сдвига мантиссы результата влево (или вправо) и вычитание (или прибавление) 1 к порядку С результата при каждом сдвиге результата до тех пор, пока будут сохраняться указанные комбинации цифр после запятой.

После нормализации суммы получаем окончательный результат:

С = А +В = М(А) 2У + М(В) 2Х = М(С) 2Р (13)

Операционный блок для выравнивания порядков при сложении и вычитании двоичных чисел с плавающей запятой представляет либо *АЛУ* порядков, либо комбинационную схему логических операций над порядками с входными и выходными регистрами.

Операции с числами с плавающей запятой производятся приближенно, так как при выравнивании порядков происходит потеря младших разрядов одного из слагаемых, а при нормализации результата вправо происходит потеря младших разрядов результата. Погрешность при этом всегда отрицательна и может достигнуть единицы младшего разряда. Чтобы уменьшить эту погрешность, применяют округление результата. Для выполнения микрооперации округления используют (*n*+1)-разрядные регистры для хранения мантисс слагаемых и (n+1)-разрядный сумматор.

На рис.5 представлена структурная схема операционного автомата, выполняющего операцию сложение двоичных чисел с плавающей запятой. Операционный автомат состоит из блоков:

1. Рг М1 – регистр мантиссы первого числа;
2. Рг М2 – регистр мантиссы второго числа;
3. Рг П1 – регистр порядка первого числа;
4. Рг П2 – регистр порядка второго числа;

# СУММАТОР1 – функциональный блок в котором осуществляется операция суммирования в дополнительном коде порядков чисел А и В;

1. СУММАТОР2 - функциональный блок в котором осуществляется операция суммирования в дополнительном коде мантисс чисел А и В;
2. Рг П – регистр порядка результата;
3. Блок анализа левой денормализации – предназначенный для анализа левой денормализации результата и осуществляющий операцию по нормализации суммы С чисел А и В, в этом случае происходит сдвиг мантиссы числа С на один разряд вправо с увеличением порядка на единицу;
4. Блок анализа правой денормализации - предназначенный для анализа правой денормализации результата и осуществляющий операцию по нормализации суммы чисел А и В, в этом случае происходит сдвиг влево мантиссы числа С до появления единицы после запятой при этом необходимо вычитать каждый раз единицу от порядка результата;

##  Рг С – регистр С результата суммы чисел А и В.

В регистр первой мантиссы Рг М1 загружается двоичный код мантиссы первого числа – А. В регистр второй мантиссы Рг М2 загружается двоичный код мантиссы второго числа – В. Регистр Рг П1 предназначен для хранения порядка первого числа. Регистр Рг П2 предназначен для хранения порядка второго числа. В СУММАТОРЕ1 осуществляется суммирование (вычитание) порядков чисел А и В в дополнительном коде. В СУММАТОРЕ2 выполняется операция суммирование –(вычитание) мантисс чисел А и В в дополнительном коде. В регистр порядка результата - Рг П записывается порядок числа С.

|  |
| --- |
| Структурная схема операционного автомата, выполняющего операцию сложение чисел с плавающей запятой. |

**Обр.К**

**Доп.К**

**Доп.К**

**Обр.К**

**В**

**Зн А Рг М1**

**Зн В Рг М2**

**Зн ПА Рг П1**

**Зн ПВ Рг П2**

# **СУММАТОР1**

# **СУММАТОР2**

**Зн П Рг П**

**Блок анализа левой**

**денормализации**

**Блок анализа правой**

**денормализации**

## **Зн С** Рг С

**А**

# Рис. 5

Блок схема алгоритма сложение чисел в форме с плавающей запятой

1

**начало**

21

19

18

16

15

14

13

12

6

5

4

3

РгП1:=РгП1 - 1

сдвиг РгМ1 влево

Зн1РгП1 **⊕**Зн2РгП1

**конец**

**1**

РгП1:=РгП1+1,

сдвиг РгМ1 вправо

РгП2:=РгП2+1, сдвиг РгМ2 вправо

РгП1<РгП2

**1**

**0**

РгМ1:=М1доп

ЗнМ1

Зн1М1 ⊕ Зн2М1

РгП1=РгП2

2

 РгП1:=Р1 РгМ1:=М1 РгП2:=Р2 РгМ2:=М2

10

9

8

7

**0**

**1**

**0**

**1**

ЗнМ2

 РгМ1:=М1доп

ЗнМ1

**0**

РгМ2:=М2доп

11

**0**

**1**

РгМ1:=РгМ1+РгМ2

РгМ1(1)

17

**0**

**1**

РгП1:=РгП1+1 , сдвиг РгМ1 вправо

**1**

**0**

**0**

**1**

20

Авар. останов

#### Рис.6

В блоках анализа левой денормализации и правой денормализации осуществляется проверка на переполнение разрядной сетки результата (левая денормализация) и единицы после запятой в прямом коде (правая денормализация).

 В регистре суммы Рг С храниться результат суммирования чисел. В регистре порядка Рг П записан порядок результата.

На рис.6 представлена блок-схема алгоритма работы операционного автомата. Обозначения, используемые в алгоритме (рис.6):

РгП1 - регистр первого порядка;

РгМ1 - регистр мантиссы первого числа;

Зн1М1 - Первый знаковый разряд мантиссы первого числа;

Зн2М1 - Второй знаковый разряд мантиссы первого числа;

 ЗнМ1 – знаковый разряд мантиссы первого числа;

 ЗнМ2 – знаковый разряд мантиссы второго числа;

РгП2 - регистр второго порядка;

РгМ2 - регистр мантиссы второго числа;

М1доп – мантисса первого числа в дополнительном коде;

М2доп – мантисса второго числа в дополнительном коде;

РгМ1(1) – первая цифра после запятой мантиссы результата;

## Работа алгоритма сложения чисел с плавающей запятой.

Блок 1 алгоритма является начальным.

В блоке 2 алгоритма осуществляется загрузка в регистры чисел А и В : в Рг П1:=Р1 – порядок первого числа, в Рг П2:=П2 – порядок второго числа, в Рг М1 – мантиссу первого числа, в Рг М2 – мантиссу второго числа.

В блоках 3, 4, 5, 6 алгоритма сформирован цикл в котором сравниваются порядки чисел. Если порядки равны блок 3 алгоритма, то осуществляется переход на блок 7. Если порядки не равны, то в блоке 4 происходит сравнение порядков РгП1 < РгП2 . В случае выполнения неравенства, в блоке 6 алгоритма происходит сложение регистра порядка первого с единицей с одновременным сдвигом регистра мантиссы первой вправо на один разряд: РгП1:=РгП1+1, сдвиг РгМ1 вправо. Процесс выравнивания будет осуществляться до тех пор, пока не будут равны порядки чисел – блок 3 алгоритма.

В случае не выполнения неравенства в блоке 4 алгоритма, то в блоке 5 алгоритма происходит сложение регистра порядка второго с единицей с одновременным сдвигом регистра мантиссы второй вправо на один разряд: РгП2:=РгП2+1, сдвиг РгМ2 вправо. Процесс выравнивания будет осуществляться до тех пор, пока не будут равны порядки чисел – блок 3 алгоритма.

В блоке 7 алгоритма анализируется знаковый разряд первой мантиссы ЗнМ1 (положительное число или отрицательное). Если знаковый разряд ЗнМ1 равен единицы, то число отрицательное и в этом случае необходимо перевести его в дополнительный код. Если знаковый разряд ЗнМ1 равен нулю, то число положительное, в этом случае переводить число в дополнительный код не нужно. Число будет представлено в прямом коде.

В блоке 8 алгоритма осуществляется перевод в дополнительный код мантиссы первого числа: РгМ1:=М1доп.

В блоке 9 алгоритма анализируется знаковый разряд второй мантиссы ЗнМ2 (положительное число или отрицательное). Если знаковый разряд ЗнМ2 равен единицы, то число отрицательное и в этом случае необходимо перевести его в дополнительный (обратный код). Если знаковый разряд ЗнМ2 равен нулю, то число положительное, в этом случае переводить число в дополнительный код не нужно. Число будет представлено в прямом коде.

В блоке 10 алгоритма осуществляется перевод в дополнительный код мантиссы второго числа: РгМ2:=М2доп.

В блоке 11 алгоритма происходит суммирование мантисс чисел А и В: РгМ1:=РгМ1+РгМ2. Результат помещается в регистр первой мантиссы – РгМ1.

В блоке 12 алгоритма по операции суммы по модулю 2 происходит сложение первого и второго знаковых разрядов мантиссы результата: Зн1М1 **⊕**Зн2М1. При сложении чисел с плавающей запятой применяются модифицированные коды. В этом случае анализируется переполнение разрядной сетки мантиссы результата и проверяется левая денормализация. Признаком переполнения разрядной сетки является комбинация двоичных знаковых разрядов 10 или 01. Самый левый разряд соответствует знаку результата.

В блоке 13 алгоритма осуществляется сдвиг мантиссы вправо на один разряд и к порядку прибавляется единица: сдвиг РгМ1 вправо, РгП1:=РгП1+1.

В блоке 14 алгоритма анализируется знаковый разряд мантиссы ЗнМ1. Если знаковый разряд равен единицы, то это означает что получено отрицательное число. В этом случае необходимо перевести результат в дополнительный код. Если ЗнМ1 равен нулю, то перевод в дополнительный код не нужен. В этом случае результат будет в прямом коде, т.е. получено положительное число.

В блоке 15 алгоритма осуществляется перевод результата в дополнительный или обратный код: РгМ1:=М1доп.

В блоке 16 алгоритма происходит анализ на правую денормализацию. Самый правый двоичный разряд мантиссы должен быть равен единицы РгМ1(1). Если условие не выполняется (выход по нулю), то осуществляется сдвиг мантиссы влево на один разряд.

В блоке 17 алгоритма происходит сдвиг мантиссы: сдвиг РгМ1 влево. Каждый сдвиг мантиссы влево сопровождается вычитанием из порядка результата единицы.

В блоке 18 алгоритма от порядка П1 вычитается единица: РгП1:=РгП1 – 1, результат помещается в регистр первого порядка. Выход этого блока поступает на вход блока 16 алгоритма. Этот процесс происходит в цикле. Выход из цикла происходит только тогда, когда появляется единица после запятой в мантиссе числа.

В блоке 19 алгоритма анализируется переполнение разрядной сетки порядка результата. Порядок числа будет представлен в модифицированном коде. Под знак отводится два двоичных разряда Зн1 и Зн2. Операция сумма по модулю 2 знаковых разрядов определяет признак переполнения разрядной сетки:

Зн1РгП1 **⊕**Зн2РгП1. Если сумма равна нулю, то происходит Аварийный останов блок 20 алгоритма. Произошло переполнение разрядной сетки. Если сумма равна единице, то происходит переход на конечный 21 блок алгоритма.

 Примеры:

1)

|  |  |  |  |
| --- | --- | --- | --- |
| А10= 12 | В10=9 | А2 =0.1100 | В2=0.1001  |
| МА=00.1100 | МВ=00.1001 | У=00.100 | Х=00.100 |
| |Δ|=00.000 | Робщ.=00.100 | Левая денорм. | Правая денорм. |
|  МСК=01.0101 |  РК=00.100 | 01. сдвиг вправо 1р | нет |
| МСПР.к=00.10101 | РПР.К=00.101 | МСК=00.10101 |  |
|  МН=00.10101 |  РН=00.101 | C2=00.10101 | С10=21 |

2)

|  |  |  |  |
| --- | --- | --- | --- |
| А10=9 | В10=-25 | А2 =0.1001 | В2= 1.11001 |
| МА=00.1001 | МВ=11.11001 | У= 00.100 | Х= 00.101 |
| МАПР.К=00.1001 | МВК=11.00111доп.к | УК =00.100 | ХК=00.101 |
| |Δ|=00.001 | Робщ.=00.101 | Левая денорм. | Правая денорм. |
|  МСК=11.10000 |  РК=00.101 | нет | нет |
| МСПР.к=11.10000 | РПР.К=00.101 | МСК=11.10000 |  |
|  МН=11.10000 |  РН=00.101 | С2=11.10000 | С10 =-16 |

3)

|  |  |  |  |
| --- | --- | --- | --- |
| А10=-10 | В10=9 | А2 =1.1010 | В2=0.1001  |
| МА=11.1010 | МВ=00.1001 | У=00.100 | Х=00.100 |
| МА.К=11.0110доп.к | МВК=00.1001 | УК =00.100 | ХК=00.101 |
| |Δ|=00.000 | Робщ.=00.100 | Левая денорм. | Правая денорм. |
|  МСК=11.1111доп.к |  РК=00.100 | нет | .0001 |
| МСПР.к=11.0001 | РПР.К=00.100 | С2=11.1 | сдвиг влево на 3р |
|  МН=11.1 |  РН=00.1 | С10=-1 | .1 |

4)

|  |  |  |  |
| --- | --- | --- | --- |
| А10=6 | В10=-0,5 | А2 =0.110 | В2=1.1  |
| МА=00.110 | МВ=11.0001 | У=00.11 | Х=11.01 |
| МАК=00.110 | МВК=11.1110об.к | УК =00.11 | ХК=11.10об |
| Δ=00.100 | Робщ.=00.11 | Левая денорм. | Правая денорм. |
|  МСК=00.1010 |  РК=00.11 | нет | нет |
| МСПР.к=00.1010 | РПР.К=00.11 |  |  |
|  =00.1011 |  РН=00.11 | С2=101,1 | С10=5,5 |

В примерах введены обозначения:

А10 , В10, С10 – числа А, В и С представлены в десятичной системе счисления;

А2, В2, С2 – числа А, В и С представлены в двоичной системе счисления;

МА, МВ – мантиссы чисел А и В в двоичной системе счисления в прямых кодах; МАК, МВК – мантиссы чисел А и В в двоичной системе счисления в дополнительном коде;

У, УК, Х, ХК – порядки чисел А и В в двоичной системе счисления в дополнительном коде;

Рк , Рпр.к , Рн– порядок результата С в прямом или дополнительном коде, нормализованный;

МСПР.к ,МСК, МН – мантисса результата С в прямом или дополнительном коде, нормализованная.

Умножение и деление чисел с плавающей запятой.

Произведение *X*·*Y* двоичных чисел с плавающей запятой *Х*=*МX*·2*px* и *Y*=*МY*·2*py* находят следующим образом:

*X*·*Y*=(*MX* ·*MY*) ·2*px*+*py*

 Мантисса произведения определяется путем умножения чисел *MX* и *MY* с фиксированной запятой, а порядок произведения является суммой порядков сомножителей.

 **Алгоритм умножения чисел с плавающей запятой:**

 1. Перемножение мантисс сомножителей как чисел с фиксированной запятой.

 2. Сложение порядков сомножителей как целых чисел с фиксированной запятой.

1. Нормализация результата, если это необходимо.
2. Округление результата.

 Частное от деления чисел с плавающей запятой *Х*=*МX*·2*px* и *Y*=*МY*·2*py* определяется следующим образом:

*X*/*Y*=*MX*/*MY*·2*px-py*

т. е. мантисса частного определяется делением мантисс *МX* и *МY* с фиксированной запятой, а порядок частного является разностью порядков делителя и делимого.

 **Алгоритм деления двоичных чисел с плавающей запятой:**

1. Деление мантисс, как целых чисел с фиксированной запятой.
2. Вычитание порядков, как целых чисел с фиксированной запятой.
3. Нормализация результата деления, если это необходимо.

 Операционные блоки деления и умножения двоичных чисел с плавающей запятой содержат операционные блоки для выполнения операций с целыми числами (мантиссами и порядками) с фиксированной запятой и характерные для операционных блоков с плавающей запятой узлы для проверки результата умножения и деления на переполнение, на нормализованность и выполнение микроопераций при необходимости нормализации результата.

**Универсальные *АЛУ* в интегральном исполнении.**

Для построения операционных частей АЛУ процессора в настоящее время используются микросхемы – универсальные АЛУ. Они представляют собой набор арифметических, логических и арифметическо-логических операций. Такие интегральные схемы имеются в комплектах ТТЛ-элементов (К155ИП), ТТЛШ-элементов (К531ИПЗП), элементов типа ЭСЛ (К500ИП181), КМДП-эдементов (564ИПЗ).

В основе построения тех и других схем АЛУ лежит применение одноразрядных комбинационных сумматоров, образующих при определенной коммутации 4-разрядный сумматор с параллельным переносом. Каждая схема АЛУ обеспечивает выполнение над 4-разрядными кодами А и В шестнадцати поразрядных логических и арифметическо-логических операций. Коммутация АЛУ на выполнение той или иной из 16 операций в заданном режиме производится управляющими сигналами. Если сигнал равен нулю, то АЛУ выполняет арифметические операции, а при единице – логические. Комбинация четырех управляющих сигналов входных сигналов определяет одну из 16 выполняемых функций в заданном режиме.

Четырехразрядные двоичные числа а3а2а1а0  и b3b2b1b0  подаются на информационные входы АЛУ соответственно, результат суммирования z3z2z1z0  фиксируется на выходах сумматора.

Сигналы переноса внутри АЛУ из младшего разряда в старший формируются схемой параллельного переноса под управлением сигнала.

Для расширения логических возможностей АЛУ используют блоки ускоренного переноса (К155ИП4, К531ИП4П, 531ИП, К500ИП179).

Для организации параллельного переноса между каскадами АЛУ с помощью блока ускоренного переноса в АЛУ предусмотрены выходы: G – выход формирования (генерации) переноса, R – выход распространения переноса и P – выход переноса со старшего разряда АЛУ.

**2. Задание**

1. Составить программу на языке высокого уровня по представленной блок-схеме алгоритма - сложение чисел с плавающей запятой в дополнительном коде рис.6;
2. Промоделировать (тестировать) программу на ПЭВМ.
3. Проанализировать результаты выполнения программы.

**3. Содержание отчета**

**Отчет должен содержать:**

* титульный лист;
* задание;
* описание переменных, используемых в программе;
* структурную схему операционного автомата;
* блок-схему алгоритма сложения чисел с плавающей запятой, представленную в дополнительном коде;
* описание работы алгоритма сложение чисел с плавающей запятой, представленного в дополнительном коде;
* текст программы;
* результаты выполнения программы.

### Контрольные вопросы

1. В каком формате представляются числа с плавающей запятой.
2. Какому диапазону принадлежит мантисса и порядок числа в формате с плавающей запятой.
3. На чем основывается принцип построения и работы операционного блока для сложения чисел с плавающей запятой двоичных чисел.
4. Поясните назначение узлов операционного блока для обработки мантисс чисел при сложении и вычитании двоичных чисел с плавающей запятой (рис. 5).
5. В какой последовательности формирует управляющие сигналы, необходимые для выполнения операции сложения с плавающей запятой операционным блоком (рис.5).
6. Какие обозначения применяются в алгоритме сложения чисел с плавающей запятой.
7. Поясните структуру и назначение каждого узла операционного блока для обработки порядков (рис. 5) при сложении и вычитании двоичных чисел с плавающей запятой.
8. Как определяется знаковый разряд результата при сложении чисел с плавающей запятой.
9. Как организован цикл сравнения порядков, а также как формируется операция сдвига вправо мантисс двоичных чисел.
10. Объясните работу блоков по определению правой и левой денормализации.
11. Как определяется признак переполнение разрядной сетки.
12. Как происходит нормализация результата чисел с плавающей запятой.
13. Какие операции при этом выполняются в операционном блоке, в алгоритме.
14. Как происходит определение признака переполнения разрядной сетки порядков при сложении чисел с плавающей запятой.
15. Как происходит округление результата.
16. Как выполняются операции умножение и деление чисел с плавающей запятой.
17. Для чего используются блоки ускоренного переноса между каскадами АЛУ.
18. Какие микросхемы – универсальные АЛУ используются в настоящее время.

**Библиографический список**

1. Карцев М.А. Арифметика цифровых машин. –М.: Наука. 1969. – 575 с.
2. Нешумова К.А. Электронные вычислительные машины и системы. – М.: Высш.шк., 1989. – 366 с.: ил.
3. Самофалов К.Г., Романкевич А.М., Валуйский В.Н. Прикладная теория цифровых автоматов. – Киев: Высш. шк., 1987 – 374 с. : ил.
4. Савельев А.Я. Прикладная теория цифровых автоматов. –М.: Высш. шк., 1987. – 271 с: ил.
5. Каган Б.М. Электронные вычислительные машины и системы. – М.: Энергия 1979. - 528 с.
6. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. – Л.: Машиностроение. 1974. – 431 с.
7. Поспелов Д.А. Арифметические основы вычислительных машин дискретного действия. – М.: Высш. шк. 1970. – 307 с.
8. Соловьев Г.Н. Арифметические устройства ЭВМ. – М.: Энергия. 1978. –177 с.
9. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники: Учеб. пособие для электротех. спец. вузов. – М.: Высш. шк., 1988. – 311 с.: ил.
10. Преснухин Л.Н., Нестеров П.В. Цифровые вычислительные машины. – М.: Высш. шк. 1981, - 511 с.