

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 02.09.2021 16:04:36
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждения высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
Локтионова
« 1 / 2 ЮЗГУ » 2018 г.



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

Методические рекомендации для лабораторной работы
для студентов укрупненной группы специальностей и
направлений подготовки 10.00.00 «Информационная безопасность»

Курск 2018

УДК 004

Составитель: С. С. Шевелёв

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Моделирование работы операционного автомата, выполняющего операцию сложение двоичных чисел с плавающей запятой. [Текст] : методические рекомендации для лабораторной работы №4 по дисциплине «Организация ЭВМ и вычислительных систем»/ Юго-Зап. гос. ун-т; сост.: С. С. Шевелёв. – Курск, 2018. – 23 с.: ил. 4, блок-схемы алгоритма. 2, табл. 4. – Библиогр.: с. 23.

Содержат сведения по вопросам работы операционного автомата, выполняющего операции деления в прямом коде. Указывается порядок выполнения практических и самостоятельных работ, правила оформления отчета.

Методические рекомендации соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальности.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 «Информационная безопасность».

Текст печатается в авторской редакции

Подписано в печать 1.02.18. Формат 60x84 1/16.
Усл.печ. л. 1,33. Уч.-изд. л. 1,21. Тираж 100 экз. Заказ. Бесплатно. 232
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

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, $A = d^{\pm p} \cdot (\pm M)$, где p -целое число, называемое порядком числа A ; d -основание системы счисления; M -мантисса числа A (обычно $|M| < 1$).

Фактически положение запятой в мантиссе M определяется величиной порядка p . С изменением p 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 $(158)_{10}$ и $(101101)_2$.

Решение. 1. $(158)_{10} = 10^3 \cdot 0,158 = 10^4 \cdot 0,0158 = 10^5 \cdot 0,00158$.

2. $(101101)_2 = 2^{110} \cdot 0,101101 = 2^{111} \cdot 0,0101101 = 2^{1000} \cdot 0,00101101$.

Числа, удовлетворяющие условию $1/d \leq M \leq 1$, 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.

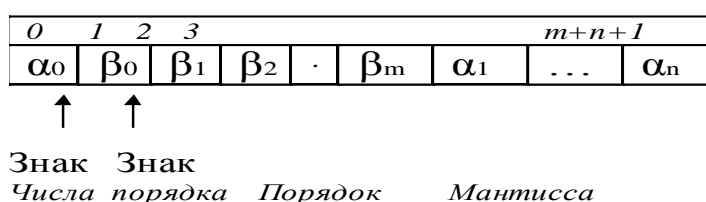


Рис.1 Разрядная сетка ЭВМ с плавающей запятой

Рис. 1

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:

$$|A|_{\max} = 2^{\overbrace{+111\dots 1}^m} 0, \overbrace{111\dots 1}^r = 2^{(2^m-1)} \cdot (1-2^{-n}) \text{ и } |A|_{\min} = 2^{\overbrace{-111\dots 1}^m} 0, \overbrace{100\dots 0}^r = 2^{-(2^m-1)} \cdot (2^{-1}) \quad (1)$$

Диапазон чисел в ЭВМ с плавающей запятой определяется неравенством

$$2^{-(2^m-1)} \cdot 2^{-1} \leq |A| \leq 2^{(2^m-1)} \cdot (1-2^{-n}) \quad (2)$$

При достаточно больших n (обычно $n \geq 30$) $1-2^{-n} \approx 1$ и неравенство принимает вид

$$2^{-2^m} \leq |A| \leq 2^{(2^m-1)} \quad (3)$$

Из этого неравенства видно, что диапазон чисел зависит в основном от порядка r . 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 A 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.

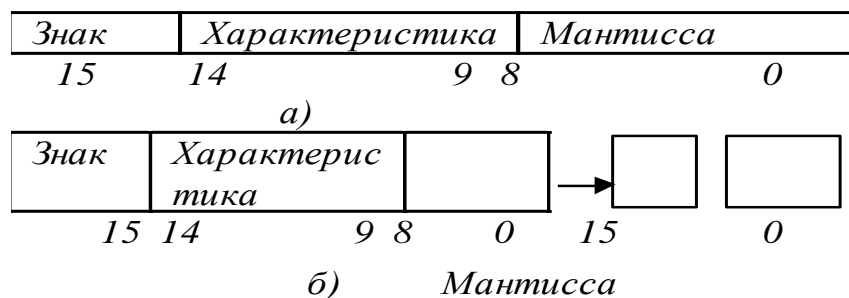


Рис.4 Форматы двоичных чисел с плавающей запятой

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 X , представленное в форме с плавающей запятой, имеет вид

$$A = M(A) P^Y, \quad (4)$$

где Y – порядок числа; $M(A)$ – мантисса; P – основание системы счисления.

Число X называется нормализованным, если его мантисса удовлетворяет условию

$$1/P \leq M(A) < 1 \quad (5)$$

или при $P=2$

$$1/2 \leq M(A) < 1 \quad (6)$$

старший разряд мантиссы всегда значащий.

Диапазон представления порядка числа лежит в пределах

$$-(2^k - 1) \leq Y \leq 2^k - 1. \quad (7)$$

Операция сложение-вычитание чисел с плавающей запятой производится 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 поместится 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 A и B . Получить сумму чисел в двоичной системе счисления.

Число A будет представлено в виде:

$$A = M_{(A)} 2^Y \quad (8)$$

Число B также будет представлено в виде:

$$B = M_{(B)} 2^X \quad (9)$$

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

$$C = A + B = M_{(A)} 2^Y + M_{(B)} 2^X \quad (10)$$

Выполнение операции будет проходить поэтапно:

1. Производится выравнивание порядков двоичных чисел Y и X . Для этого необходимо определить порядок какого числа меньше. Операция сравнения порядков осуществляется на сумматоре, на котором из порядка первого числа Y вычитают порядок второго числа X . 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 Y число X . Все операции выполняются на сумматоре, при этом число X будет представлено в дополнительном (обратном) коде. Если разность порядков Δ положительная, то необходимо сдвигать вправо мантиссу числа X . Если разность порядков Δ отрицательная, то сдвигать вправо надо мантиссу числа Y . Если $\Delta > 0$, то при каждом сдвиге из разности порядков необходимо вычитать 1, а если $\Delta < 0$, 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 $M'_{(B)}$.

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:

$$C = A + B = M_{(A)} 2^Y + M_{(B)} 2^X = M_{(A)} 2^{P_{\text{общ.}}} + M'_{(B)} 2^{P_{\text{общ.}}} \quad (11)$$

выносим за скобки общий порядок чисел - $2^{P_{\text{общ.}}}$. Получаем:

$$C = 2^{P_{\text{общ.}}} (M_{(A)} + M'_{(B)}) = 2^{P_{\text{общ.}}} M_{(C)} \quad (12)$$

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

После нормализации суммы получаем окончательный результат:

$$C = A + B = M_{(A)} 2^Y + M_{(B)} 2^X = M_{(C)} 2^P \quad (13)$$

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;
5. СУММАТОР1 – функциональный блок, в котором осуществляется операция суммирования в дополнительном коде порядков чисел А и В;
6. СУММАТОР2 - функциональный блок, в котором осуществляется операция суммирования в дополнительном коде мантисс чисел А и В;
7. Рг П – регистр порядка результата;
8. 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;
9. 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;
10. 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.

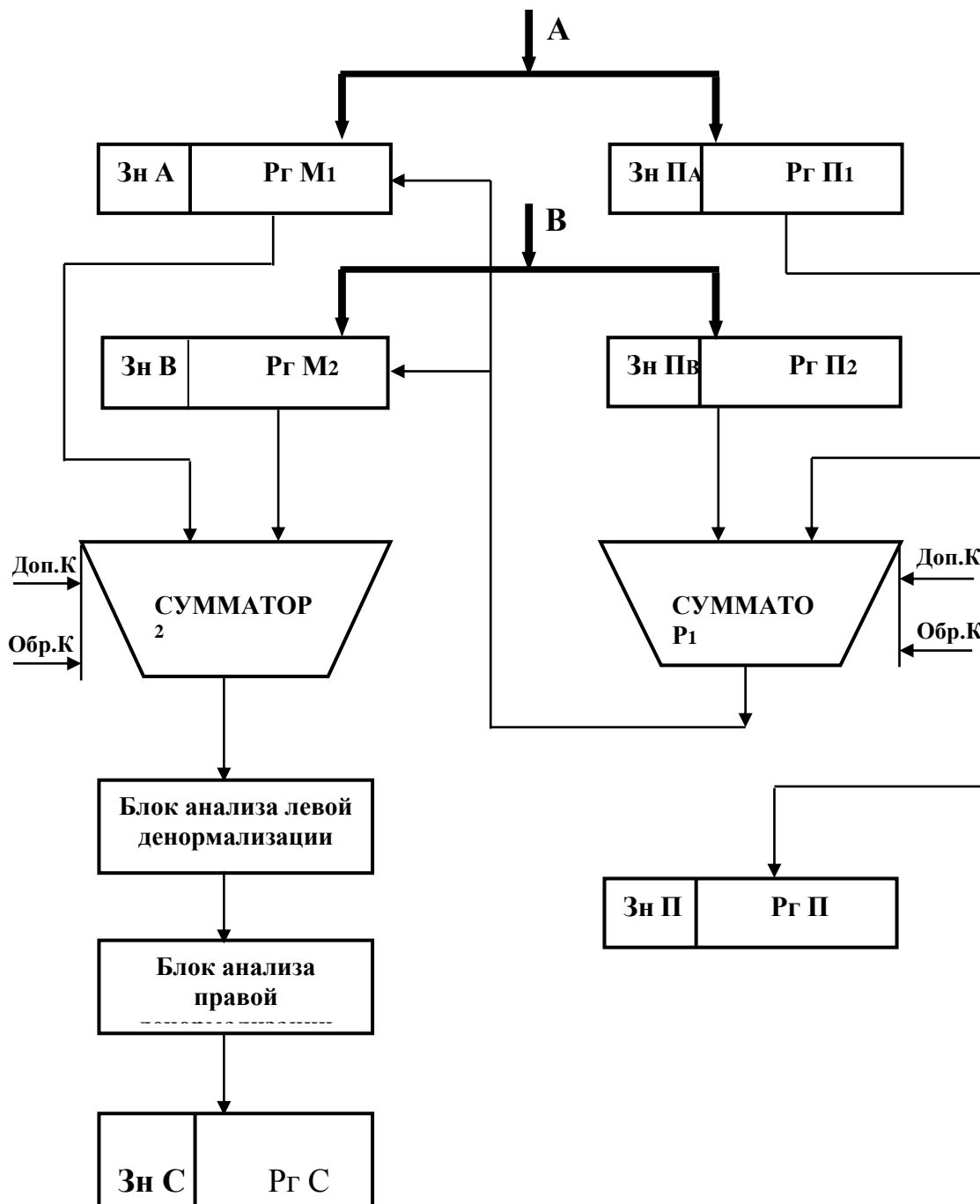
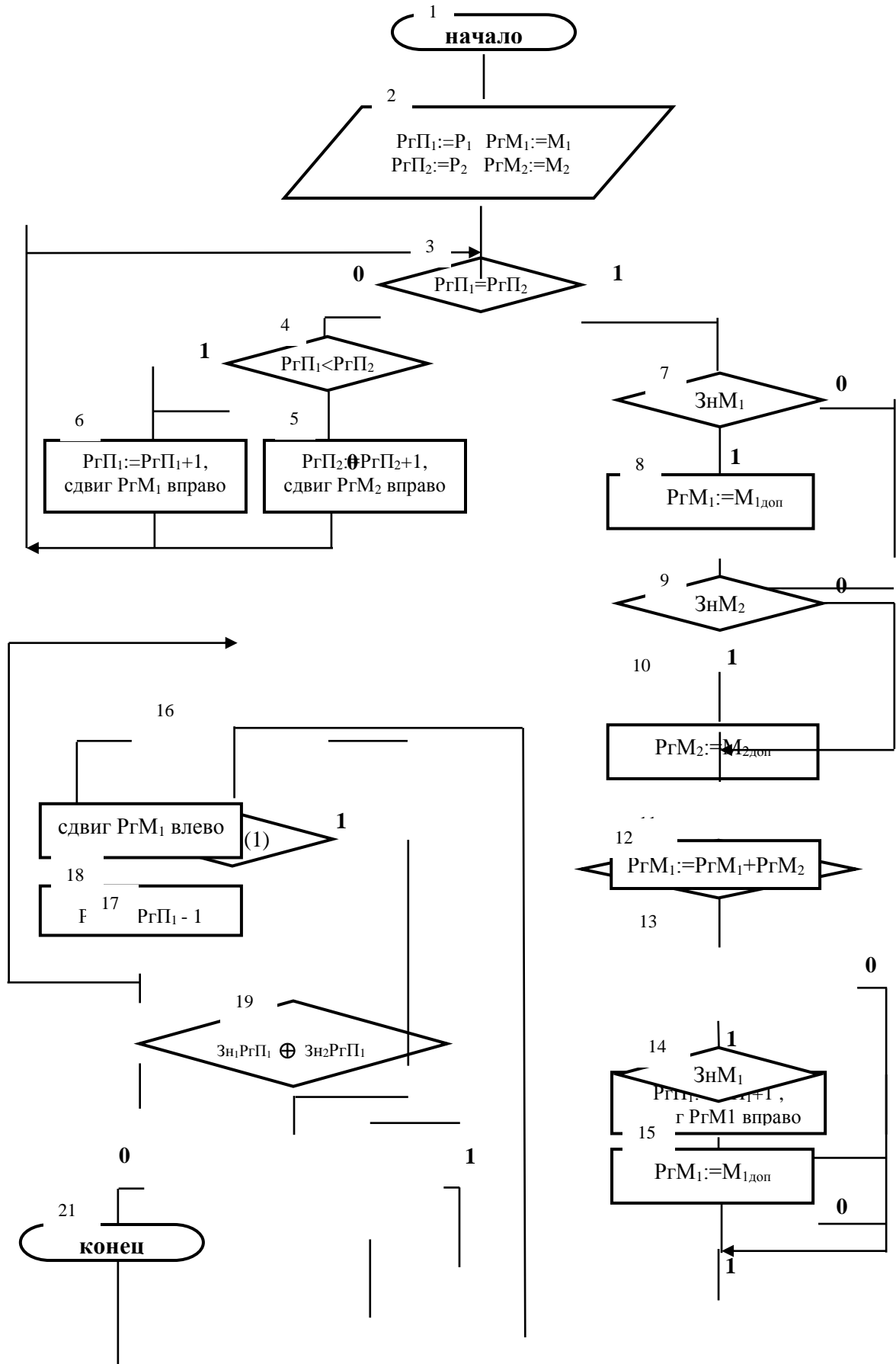


Рис. 5

Блок схема алгоритма сложение чисел в форме с плавающей запятой



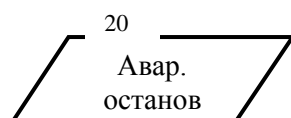


Рис. 6

В блоках анализа Рис.6 нормализации 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 мантииссы первого числа;
 - Зн₁М₁ - Первый знаковый разряд мантииссы первого числа;
 - Зн₂М₁ - Второй знаковый разряд мантииссы первого числа;
 - ЗнМ₁ – знаковый разряд мантииссы первого числа;
 - ЗнМ₂ – знаковый разряд мантииссы второго числа;
 - РгП₂ - регистр второго порядка;
 - РгМ₂ - регистр мантииссы второго числа;
 - М_{1доп} – мантиисса первого числа в дополнительном коде;
 - М_{2доп} – мантиисса второго числа в дополнительном коде;
 - РгМ₁(1) – первая цифра после запятой мантииссы 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 – мантииссу первого числа, в Рг М2 – мантииссу 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 $РгП_1 < РгП_2$. В случае

выполнения неравенства, в блоке 6 алгоритма происходит сложение регистра порядка первого с единицей с одновременным сдвигом регистра мантииссы первой вправо на один разряд: $RгП_1 := RгП_1 + 1$, сдвиг $RгM_1$ 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 мантииссы второй вправо на один разряд: $RгП_2 := RгП_2 + 1$, сдвиг $RгM_2$ 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 мантииссы $ЗнM_1$ (положительное число или отрицательное). Если знаковый разряд $ЗнM_1$ равен единицы, то число отрицательное и в этом случае необходимо перевести его в дополнительный код. Если знаковый разряд $ЗнM_1$ 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 мантииссы первого числа: $RгM_1 := M_{1\text{доп}}$.

В блоке 9 алгоритма анализируется знаковый разряд второй мантииссы $ЗнM_2$ (положительное число или отрицательное). Если знаковый разряд $ЗнM_2$ равен единицы, то число отрицательное и в этом случае необходимо перевести его в дополнительный (обратный код). Если знаковый разряд $ЗнM_2$ 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 мантииссы второго числа: $RгM_2 := M_{2\text{доп}}$.

В блоке 11 алгоритма происходит суммирование мантиисс чисел А и В: $RгM_1 := RгM_1 + RгM_2$. Результат помещается в регистр первой мантииссы – $RгM_1$.

В блоке 12 алгоритма по операции суммы по модулю 2 происходит сложение первого и второго знаковых разрядов мантииссы результата: $Зн_1M_1 \oplus Зн_2M_1$. При сложении чисел с плавающей запятой применяются модифицированные коды. В этом случае анализируется переполнение разрядной сетки мантииссы 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 мантиисы вправо на один разряд и к порядку прибавляется единица: сдвиг $RгM_1$ вправо, $RгП_1 := RгП_1 + 1$.

В блоке 14 алгоритма анализируется знаковый разряд мантиисы $ЗнM_1$. Если знаковый разряд равен единицы, то это означает что получено отрицательное число. В этом случае необходимо перевести результат в дополнительный код. Если $ЗнM_1$ 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: $RгM_1 := M_{1\text{доп}}$.

В блоке 16 алгоритма происходит анализ на правую денормализацию. Самый правый двоичный разряд мантиисы должен быть равен единицы $RгM_1(1)$. Если условие не выполняется (выход по нулю), то осуществляется сдвиг мантиисы влево на один разряд.

В блоке 17 алгоритма происходит сдвиг мантиисы: сдвиг $RгM_1$ влево. Каждый сдвиг мантиисы влево сопровождается вычитанием из порядка результата единицы.

В блоке 18 алгоритма от порядка $П_1$ вычитается единица: $RгП_1 := RгП_1 - 1$, 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 мантиисе числа.

В блоке 19 алгоритма анализируется переполнение разрядной сетки порядка результата. Порядок числа будет представлен в модифицированном коде. Под знак отводится два двоичных разряда $Зн_1$ и $Зн_2$. Операция сумма по модулю 2 знаковых разрядов определяет признак переполнения разрядной сетки:

$Зн_1 RгП_1 \oplus Зн_2 RгП_1$. 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)

	$A_{10}=12$	$B_{10}=9$	A_2 $=0.1100$	$B_2=0.10$ 01
100	$M_A=00.1$	$M_B=00.10$	$Y=00.100$	$X=00.1$ 00
00	$ \Delta =00.0$	$P_{\text{общ.}}=00.1$	Левая денорм.	Правая денорм.
$M_{CK}=01.0101$	$P_K=00.100$	01. сдвиг вправо 1p	нет	
$M_{CIP.K}=0.10101$	$P_{IP.K}=00.101$	$M_{CK}=00.10101$		
$M_H=00.10101$	$P_H=00.101$	$C_2=00.10$ 101	$C_{10}=21$	

2)

	$A_{10}=9$	$B_{10}=-25$	A_2 $=0.1001$	$B_2=$ 1.11001
001	$M_A=00.1$	$M_B=11.11$	$Y=$ 00.100	$X=$ 00.101
00.1001	$M_{AIP.K}=$	$M_{BK}=11.0$ 0111 _{доп.к}	Y_K $=00.100$	$X_K=00.1$ 01
01	$ \Delta =00.0$	$P_{\text{общ.}}=00.1$	Левая денорм.	Правая денорм.
$M_{CK}=11.10000$	$P_K=00.101$	нет	нет	
$M_{CIP.K}=1.10000$	$P_{IP.K}=00.101$	$M_{CK}=11.10000$		
$M_H=11.10000$	$P_H=00.101$	$C_2=11.1$ 0000	$C_{10}=-16$	

3)

	$A_{10}=-10$	$B_{10}=9$	A_2 $=1.1010$	$B_2=0.100$ 1
010	$M_A=11.1$	$M_B=00.10$	$Y=00.10$ 0	$X=00.10$ 0
$M_{A.K}=11.$	$M_{BK}=00.1$	Y_K	$X_K=00.1$	

0110 _{доп.к}	001	=00.100	01
00	$ \Delta =00.0$	$P_{общ.}=00.1$	Левая денорм.
$M_{СК}=11.1111_{доп.к}$	$P_K=00.100$	нет	Правая денорм.
1.0001	$M_{СПР.к}=1$	$P_{ПР.к}=00.$.0001
$M_H=11.1$	100	$C_2=11.1$	сдвиг влево на 3p
	$P_H=00.1$	$C_{10}=-1$.1

4)

$A_{10}=6$	$B_{10}=-0,5$	A_2	$B_2=1.1$
10	$M_A=00.1$	=0.110	$M_B=11.00$
110	01	$Y=00.11$	$X=11.01$
0	$M_{AK}=00.$	U_K	$M_{BK}=11.1$
	$\Delta=00.10$	=00.11	$0_{об}$
$M_{СК}=00.1010$	1	Левая денорм.	$P_{общ.}=00.1$
0.1010	$P_K=00.11$	нет	Правая денорм.
=00.1011	$M_{СПР.к}=0$		нет
	11	$C_2=101,$	$C_{10}=5,5$
	$P_H=00.11$	1	

В примерах введены обозначения:

A_{10} , B_{10} , C_{10} – числа A, B и C представлены в десятичной системе счисления;

A_2 , B_2 , C_2 – числа A, B и C представлены в двоичной системе счисления;

M_A , M_B – мантиссы чисел A и B в двоичной системе счисления в прямых кодах; M_{AK} , M_{BK} – мантиссы чисел A и B в двоичной системе счисления в дополнительном коде;

U , U_K , X , X_K – порядки чисел A и B в двоичной системе счисления в дополнительном коде;

P_k , $P_{пр.k}$, P_n – порядок результата C в прямом или дополнительном коде, нормализованный;

$M_{спр.k}$, $M_{ск}$, M_n – мантисса результата C в прямом или дополнительном коде, нормализованная.

Умножение и деление чисел с плавающей запятой.

Произведение $X \cdot Y$ двоичных чисел с плавающей запятой $X = M_X \cdot 2^{p_x}$ и $Y = M_Y \cdot 2^{p_y}$ находят следующим образом:

$$X \cdot Y = (M_X \cdot M_Y) \cdot 2^{p_x + p_y}$$

Мантисса произведения определяется путем умножения чисел M_X и M_Y 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.
3. Нормализация результата, если это необходимо.
4. Округление результата.

Частное от деления чисел с плавающей запятой $X = M_X \cdot 2^{p_x}$ и $Y = M_Y \cdot 2^{p_y}$ определяется следующим образом:

$$X/Y = M_X/M_Y \cdot 2^{p_x - p_y}$$

т. е. мантисса частного определяется делением мантисс M_X и M_Y 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 арифметико-логических операций. Такие интегральные схемы имеются в комплектах ТТЛ-элементов (К155ИП), ТТЛШ-элементов (К531ИПЗП), элементов типа ЭСЛ (К500ИП181), КМДП-элементов (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 арифметико-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 $a_3a_2a_1a_0$ и $b_3b_2b_1b_0$ подаются на информационные входы АЛУ соответственно, результат суммирования $z_3z_2z_1z_0$ 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 сложения 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. Пospelов 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.