МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

УЛЬЯНОВСК 2003

Министерство образования Российской Федерации

Ульяновский государственный технический университет

АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

Методические указания к практическим занятиям по курсу «Прикладная теория информации»

Составители: В.Д. Горбоконенко, В.Е. Шикина

Ульяновск 2003

УДК 621.3+681.3

Рецензент канд. техн. наук, доцент кафедры ТОР УлГТУ Б.Н. Романов

Одобрено секцией методических пособий научно-методического совета Ульяновского государственного технического университета

Арифметические основы цифровой техники: Методические указания к практическим занятиям по курсу «Прикладная теория информации»/ Сост.: В.Д. Горбоконенко, В.Е. Шикина. – Ульяновск: УлГТУ, 2003. – 27 с.

Указания написаны в соответствии с рабочей программой курса «Прикладная теория информации» для специальностей 071900 «Информационные системы в технике и технологиях» и 1903300 «АП и ИВК». В них содержатся математические основы информатики, включающие системы счисления и формы представления информации.

Приведенный материал может быть использован студентами при подготовке к практическим занятиям. Методические указания подготовлены на кафедре «Измерительно-вычислительные комплексы».

УДК 621.3+681.3

ОГЛАВЛЕНИЕ

1. СИСТЕМЫ СЧИСЛЕНИЯ

1.1. Представление чисел в различных системах счисления

Под системой счисления понимают совокупность правил изображения числа с помощью цифровых знаков. Если при записи числа значимость цифры зависит от места (позиции), занимаемого в числе, то система счисления называется *позиционной*. Для представления чисел в цифровых устройствах, а также для представления разнообразной информации в процессе программирования наряду с привычной для нас десятичной системой счисления широко используются другие. Рассмотрим принцип построения наиболее употребительных позиционных систем счисления.

Числа в таких системах счисления представляются последовательностью цифр (цифр разрядов), разделенных запятой на две группы: группу разрядов, изображающую целую часть числа, и группу разрядов, изображающую дробную часть числа:

$$\ldots a_2 a_1 a_0,\ a_{-1} a_{-2} \ldots$$

Здесь a_0, a_1… обозначают цифры нулевого, первого и т.д. разрядов целой части числа; a_{-1}, a_{-2}… – цифры первого, второго и т.д. разрядов дробной части числа..

Единице каждого разряда приписан определенный вес p^k, где p — основание системы счисления, k – номер разряда, равный индексу при буквах, изображающих цифры разрядов. Так, представленная запись означает следующее количество:

$$N = \ldots + a_2 \cdot p^2 + a_1 \cdot p^1 + a_0 \cdot p^0 + a_{-1} \cdot p^{-1} + a_{-2} \cdot p^{-2} + \ldots$$

Для представления цифр разрядов используется набор из p различных символов. Так, при $p = 10$ (т. е. в обычной десятичной системе счисления) для записи цифр разрядов используется набор из десяти символов: 0, 1, 2, ..., 9. При этом запись числа $729{,}324_{10}$ (здесь и в дальнейшем индекс при числе будет указывать основание системы счисления, в которой представлено число) означает следующее количество:

$$\begin{array}{cccccccc} 7 & 2 & 9 & , & 3 & 2 & 4_{10} & = 7 \cdot 10^2 + 2 \cdot 10^1 + 9 \cdot 10^0 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2} + 4 \cdot 10^{-3} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \\ 10^2 & 10^1 & 10^0 & & 10^{-1} & 10^{-2} & 10^{-3} & \end{array}$$

Весовые коэффициенты разрядов

Используя такой принцип представления чисел, но выбирая различные значения основания *p,* можно строить разнообразные системы счисления.

Двоичная система счисления. Основание системы счисления p=2. Таким образом, для записи цифр разрядов требуется набор всего лишь из двух символов, в качестве которых используются 0 и 1. Следовательно, в двоичной системе счисления число представляется последовательностью символов 0 и 1. При этом запись $11011{,}101_2$ соответствует в десятичной системе счислении следующему числу:

$$\begin{array}{cccccccc} 1 & 1 & 0 & 1 & 1, & 1 & 0 & 1_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} \end{array} = (1\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 + + 1\cdot2^{-1} + 0\cdot2^{-2} + 1\cdot2^{-3})_{10} = 27,625_{10}.$$

Весовые коэффициенты разрядов

Восьмеричная система счисления. Основание системы p=8. Следовательно, для представления цифр разрядов должно использоваться восемь разных символов, в качестве которых выбраны 0, 1, 2,…, 7. Символы 8 и 9 здесь не используются и в записи чисел встречаться не должны. Например, записи $735{,}46_8$ в десятичной системе счисления соответствует следующее число:

$$\begin{array}{cccccc} 7 & 3 & 5 & , & 4 & 6_8 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 8^2 & 8^1 & 8^0 & & 8^{-1} & 8^{-2} \end{array} = (7\cdot8^2 + 3\cdot8^1 + 5\cdot8^0 + 4\cdot8^{-1} + 6\cdot8^{-2})_{10} = 477,59375_{10}.$$

Весовые коэффициенты разрядов

Шестнадцатеричная система счисления. Основание системы счисления p=16 и для записи цифр разрядов должен использоваться набор из шестнадцати символов: 0, 1, 2,..., 9, A, B, C, D, E, F. В нем используются десять арабских цифр, и до требуемых шестнадцати их дополняют шестью начальными буквами латинского алфавита. При этом символ A соответствует количеству, в десятичной системе счисления равному 10, B – 11, C – 12, D – 13, E – 14 и F – 15.

Запись $AB9{,}C2F_{16}$ соответствует следующему числу в десятичной системе счисления:

$$\begin{array}{cccccc} A & B & 9, & C & 2 & F_{16} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 16^2 & 16^1 & 16^0 & 16^{-1} & 16^{-2} & 16^{-3} \end{array} = (10\cdot16^2 + 11\cdot16^1 + 9\cdot16^0 + 12\cdot16^{-1} + + 2\cdot16^{-2} + 15\cdot16^{-3})_{10} = 2745,7614745\ldots_{10}$$

Весовые коэффициенты разрядов

Хранение n-разрядных чисел в цифровой аппаратуре можно осуществить с помощью устройств, содержащих n элементов, каждый из которых запоминает цифру соответствующего разряда числа. Наиболее просто осуществляется хранение чисел, представленных в двоичной системе счисления. Для запоминания цифры каждого разряда двоичного числа могут использоваться устройства с двумя устойчивыми состояниями (триггеры). Одному из этих устойчивых состояний ставится в соответствие цифра 0, другому – цифра 1.

При хранении десятичных чисел каждая цифра десятичного числа предварительно представляется в двоичной форме. Такая форма представления чисел называется *двоично-кодированной десятичной системой*. Например, число $765{,}93_{10}$ в двоично-кодированной десятичной системе представляется в следующем виде:

$$765,93_{10} = \begin{array}{ccccc} 0111 & 0110 & 0101\;, & 1001 & 0011_{2\text{-}10} \\ \mathbf{7} & \mathbf{6} & \mathbf{5} & \mathbf{9} & \mathbf{3} \end{array}.$$

Заметим, что, несмотря на внешнее сходство двоично-кодированного десятичного числа с двоичным числом, оно не является двоичным. В этом можно легко убедиться. Если целую часть приведенной выше записи в правой части равенства рассматривать как двоичное число, то оно при переводе в десятичную систему счисления означало бы 1893_{10}, что не совпадает с целой частью исходного числа 765.

Рассмотренный способ двоичного представления (кодирования) десятичных цифр использует так называемый *код 8421* (название кода составлено из весовых коэффициентов разрядов в двоичной системе счисления).

1.2. Преобразование чисел из одной системы счисления в другую

Основания восьмеричной и шестнадцатеричной систем счисления выражаются целой степенью двух ($8=2^3$; $16=2^4$). Этим объясняется простота преобразования чисел между этими системами и двоичной системой счисления.

Для перевода чисел из восьмеричной системы счисления в двоичную достаточно каждую цифру восьмеричного числа заменить соответствующим трехразрядным двоичным числом. Например,

$$762,35_8 = \underset{\mathbf{7}}{111}\ \underset{\mathbf{6}}{110}\ \underset{\mathbf{2}}{010}\ ,\ \underset{\mathbf{3}}{011}\ \underset{\mathbf{5}}{101}\,_{2.}$$

Перевод в двоичную систему счисления шестнадцатеричных чисел достигается заменой цифр шестнадцатеричного представления четырехразрядными двоичными числами. Например,

$$A7B,C7_{16} = \underset{\mathbf{A}}{1010}\ \underset{\mathbf{7}}{0111}\ \underset{\mathbf{B}}{1011}\ ,\ \underset{\mathbf{C}}{1100}\ \underset{\mathbf{7}}{0111}\,_{2.}$$

При обратном переводе чисел из двоичной системы в восьмеричную или шестнадцатеричную систему счисления необходимо разряды двоичного числа, отсчитывая их от запятой влево и вправо, разбить на группы по три разряда (в случае перевода в восьмеричную систему) или на группы по четыре разряда (в случае перевода в шестнадцатеричную систему счисления). Неполные крайние группы дополняются до полных нулями. Затем каждая двоичная группа представляется цифрой той системы счисления, в которую переводится число. Например,

$$\underset{\mathbf{1}}{001}\ \underset{\mathbf{7}}{111}\ ,\ \underset{\mathbf{5}}{101}\ \underset{\mathbf{2}}{010}\,_2 = 17,52_8; \qquad \underset{\mathbf{5}}{0101}\ \underset{\mathbf{C}}{1100}\ ,\ \underset{\mathbf{D}}{1101}\ \underset{\mathbf{6}}{0110}\,_2 = 5C,D6_{16.}$$

Большую сложность представляет перевод чисел между десятичной и двоичной системами счисления. Метод, используемый для такого перевода, зависит от того, в какой системе счисления представлены числа, над которыми проводятся необходимые для перевода чисел арифметические операции. Если перевод производится человеком, то, очевидно, операции будут выполняться над числами в десятичной системе счисления; если же перевод осуществляется цифровым устройством, арифметические операции удобнее выполнять над числами в двоичной системе счисления. Рассмотрим эти два случая.

Перевод чисел с выполнением операций над десятичными числами. Так как преобразование чисел между двоичной и шестнадцатеричной системами счисления не представляет труда, то для простоты выкладок будем в дальнейшем рассматривать перевод чисел между шестнадцатеричной и десятичной системами счислении.

Пусть требуется перевести число из шестнадцатеричной в десятичную систему счисления. В качестве примера выберем число $9A5F,C83B_{16}$. С учетом весов разрядов шестнадцатеричной системы счисления запишем это число в десятичной системе счисления:

$$9A5F,C83B_{16} =$$

$$= \underbrace{(9\cdot 16^3 + \overset{\substack{A\\ \uparrow}}{10\cdot 16^2} + 5\cdot 16^1 + \overset{\substack{F\\ \uparrow}}{15\cdot 16^0}}_{\text{Целая часть}} \underbrace{+ \overset{\substack{C\\ \uparrow}}{12\cdot 16^{-1}} + 8\cdot 16^{-2} + 3\cdot 16^{-3} + \overset{\substack{B\\ \uparrow}}{11\cdot 16^{-4}}}_{\text{Дробная часть}})_{10} =$$

$$= \Big(\underbrace{((9\cdot 16 + 10)\cdot 16 + 5)\cdot 16 + 15}_{\text{Целая часть}} + \underbrace{16^{-1}\cdot\big(12 + 16^{-1}\cdot(8 + 16^{-1}\cdot(3 + 16^{-1}\cdot 11))\big)}_{\text{Дробная часть}}\Big)_{10}.$$

Здесь путем группировки членов вычисление полиномов представлено в форме так называемой *схемы Горнера*, удобной для программирования и обеспечивающей минимальное число выполняемых операций умножения.

Вычисления в приведенном примере дают следующий результат:

$$9A5F,C83B_{16} = 39519,7821502_{10}.$$

Целая часть числа преобразуется точно, дробная часть — приближенно. В приведенном примере вычисления при нахождении дробной части выполнялись с точностью, определяемой семью десятичными разрядами.

Рассмотрим обратный перевод чисел из десятичной в шестнадцатеричную систему счисления. Воспользуемся приведенным выше примером. Теперь будем считать заданным десятичное число $39519,7821502_{10}$ и будем искать его представление в шестнадцатеричной системе счисления. Преобразуем целую часть числа. Из равенства

$$39519_{10} = ((9\cdot 16+\underset{\substack{\uparrow\\ A}}{10})\cdot 16+5)\cdot 16+\underset{\substack{\uparrow\\ F}}{15}$$

можно вывести следующее правило получения цифр шестнадцатеричного представления. Деление правой части равенства (т. е. целой части заданного числа) на 16 дает в частном $(9\cdot 16+10)\cdot 16+5$ и в остатке 15 (т. е. F); деление полученного частного на 16 даст частное $9\cdot 16+10$ и остаток 5; деление последнего частного приведет к частному 9 и остатку 10 (т. е. A). Таким образом, последовательно деля на 16 целую часть десятичного числа и образующиеся частные, получаем в последнем частном и остатках цифры всех разрядов шестнадцатеричного представления целой части числа. Покажем эти действия по преобразованию десятичного числа 39519_{10} в шестнадцатеричную систему счисления:

$$\begin{array}{llllll}
-\,39519 & \underline{|\,16\qquad} & & & & \\
\underline{39504} & -\,2469 & \underline{|\,16\qquad} & & & \\
\quad 15 & \underline{\;2464\;} & -\,154 & \underline{|\,16\qquad} & & \\
\quad\uparrow & \qquad 5 & \underline{\;144\;} & \qquad 9 & & \\
\quad F & & \;\;10 & & & \\
 & & \;\;\uparrow & & & \\
 & & \;\;A & & &
\end{array}$$

Отсюда $39519_{10}=9A5F_{16}$.

Теперь рассмотрим преобразование дробной части десятичного числа в шестнадцатеричную систему счисления. Из равенства

$$0,7821502_{10} = 16^{-1}\cdot\left(\underset{\substack{\uparrow\\ C}}{12}+16^{-1}\cdot(8+16^{-1}\cdot(3+16^{-1}\cdot\underset{\substack{\uparrow\\ B}}{11}))\right)$$

следует, что для получения цифр разрядов дробной части шестнадцатеричного числа ($0,C83B_{16}$) необходимо последовательно умножать на 16 дробную часть исходного десятичного числа и дробные части образующихся произведений. При этом целые части этих произведений являются цифрами шестнадцатеричного представления:

$$\begin{array}{lcr}
 & & 0,7821502 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
C & \longleftarrow & 12,5144032 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
8 & \longleftarrow & 8,2304512 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
3 & \longleftarrow & 3,6872192 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
A & \longleftarrow & 10,9955072 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
F & \longleftarrow & 15,9281152 \\
 & & \times\qquad\quad \\
 & & \underline{\qquad\qquad 16} \\
E & \longleftarrow & 14,8498432 \\
 & & \ldots\ldots\ldots\ldots
\end{array}$$

Таким образом, $0,7821502_{10} = 0,C83AFE\ldots_{16} \approx 0,C83B_{16}$. И в этом случае убеждаемся, что дробные числа преобразуются неточно.

Перевод чисел с выполнением операций в двоичной системе счисления. Рассмотрим перевод десятичных чисел в двоичную систему счисления. Для иллюстрации метода перевода выберем десятичное число $937,568_{10}$, которое представим в следующей форме:

$$937,568_{10} = (9\cdot10^2 + 3\cdot10^1 + 7\cdot10^0 + 5\cdot10^{-1} + 6\cdot10^{-2} + 8\cdot10^{-3})_{10} =$$
$$= \underbrace{(9\cdot10 + 3)\cdot10 + 7}_{\text{Целая часть}} + \underbrace{10^{-1}\cdot(5 + 10^{-1}\cdot(6 + 10^{-1}\cdot8))}_{\text{Дробная часть}}.$$

Представив числа, входящие в правую часть равенства, четырехразрядными двоичными числами, запишем выражения для преобразования целой и дробной частей:

$$937_{10} = \big((\underset{\mathbf{9}}{1001} \cdot \underset{\mathbf{10}}{1010} + \underset{\mathbf{3}}{0011}) \cdot \underset{\mathbf{10}}{1010} + \underset{\mathbf{7}}{0111}\big)_2,$$

$$0,568_{10} = \big(((\underset{\mathbf{8}}{1000} : \underset{\mathbf{10}}{1010} + \underset{\mathbf{6}}{0110}) : \underset{\mathbf{10}}{1010} + \underset{\mathbf{5}}{0101}) : \underset{\mathbf{10}}{1010}\big)_2.$$

Получаемые в результате выполнения операций над двоичными числами значения представляют собой двоичные представления соответственно целой и дробной частей исходного числа.

Рассмотрим обратный перевод двоичных чисел в десятичную систему счисления. Перевод целых двоичных чисел производится последовательным делением в двоичной системе счисления на число 1010_2 исходного двоичного и всех образующихся частных. При этом последнее частное и возникающие при делении остатки являются двоичными представлениями цифр разрядов искомого десятичного представления числа.

Перевод дробного двоичного числа производится последовательным умножением на двоичное число 1010_2 исходного числа и дробных частей получаемых произведений. При этом целые части произведений являются двоичным представлением цифр разрядов искомого десятичного представления дробного числа.

Преобразование чисел с помощью сдвиговых регистров. Рассмотрим преобразование двоичных чисел в десятичную систему счисления. Пусть число, подлежащее преобразованию в десятичную систему счисления, хранится в регистре R_1 (рис.1.1). Результат преобразования (число в десятичной системе счисления) будем формировать в регистре R_2. Разряды регистра R_2 делятся на 4-разрядные группы R_2', R_2'', R_2''' и т.д., каждая из которых предназначена для хранения одной десятичной цифры, представленной в двоичной системе счисления.

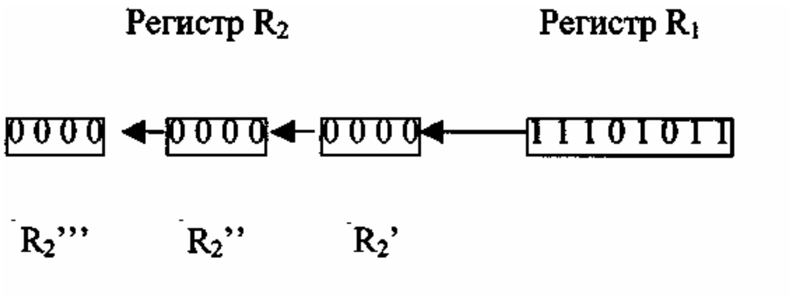


Рис. 1.1

Рассматриваемый способ преобразования потребует выполнения последовательности операций сдвига влево содержимого регистров R_1 и R_2 с передачей выдвигаемого из регистра R_1 содержимого старшего разряда в освобождающийся младший разряд регистра R_2. Двоичное число, выдвигаясь из регистра R_1, будет вдвигаться в регистр R_2. При этом необходимо учитывать следующую особенность выполнения сдвигов в регистре R_2. Единица, выдвигаемая при сдвиге из старшего разряда группы R_2', имеет вес $2^4 = 16$. Однако, поступая в группу R_2'' (в разряд десятков), эта единица будет иметь вес 10. Таким образом, при передаче единицы из R_2' в R_2'' происходит потеря 6 единиц. Для компенсации этой потери потребуется прибавить 6 единиц к содержимому R_2'. Можно показать, что выдвижение единицы из любой 4-разрядной группы регистра R_2 требует коррекции содержимого этой группы путем прибавления 6 единиц. Такая же коррекция требуется и в случае, когда после сдвига в 4-разрядной группе возникает число, большее или равное 10. В этом случае прибавление 6 единиц вызывает перенос из старшего разряда группы, который необходимо прибавлять к содержимому следующей 4-разрядной группы.

Более удобным оказывается способ, при котором коррекция производится не после сдвига, а до выполнения сдвига влево. В этом случае коррекция осуществляется прибавлением числа 3 (в результате сдвига оно удваивается и принимает значение 6), а признаком необходимости коррекции является наличие в 4-разрядной группе числа, большего или равного 5 (после сдвига это число оказывается большим или равным 10). При этом передача единицы в следующую 4-разрядную группу осуществляется только путем передачи переноса, возникающего в процессе сдвига (т. е. исключается необходимость прибавления единицы к содержимому группы, как в случае, когда коррекция выполняется после операции сдвига).

Если число разрядов регистра R_1 равно n, то преобразование завершается после n-кратного выполнения сдвига.

В таблице 1.1 показан процесс преобразования числа $N = 11101011_2$ десятичное представление 235_{10}.

Таблица 1.1

R_2'''	R_2''	R_2'	R_1	
0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0 1 0 1 1	Исходное состояние
0 0 0 0	0 0 0 0	0 0 0 1	1 1 0 1 0 1 1	Сдвиг
0 0 0 0	0 0 0 0	0 0 1 1	1 0 1 0 1 1	Сдвиг
0 0 0 0	0 0 0 0	+0 1 1 1	0 1 0 1 1	Сдвиг
		0 0 1 1		Коррекция
		1 0 1 0		
0 0 0 0	0 0 0 1	0 1 0 0	1 0 1 1	Сдвиг
0 0 0 0	0 0 1 0	+1 0 0 1	0 1 1	Сдвиг
		0 0 1 1		Коррекция
		1 1 0 0		
0 0 0 0	+0 1 0 1	+1 0 0 0	1 1	Сдвиг
	0 0 1 1	0 0 1 1		Коррекция
	1 0 0 0	1 0 1 1		
0 0 0 1	0 0 0 1	+0 1 1 1	1	Сдвиг
		0 0 1 1		Коррекция
		1 0 1 0		
0 0 1 0	0 0 1 1	0 1 0 1		Сдвиг
2	**3**	**5**		

Рассмотрим обратное преобразование числа из десятичной системы счисления в двоичную. Очевидно, такое преобразование может быть осуществлено при использовании описанных выше действий, выполняемых в обратном порядке: осуществляется серия сдвигов вправо содержимого регистров R_2 и R_1 с коррекцией результата после каждого сдвига. Коррекция выполняется путем вычитания трех единиц, если содержимое 4-разрядной группы окажется больше или равно 8. Поясним необходимость в такой коррекции. Указанное условие выполняется, если происходит передача единицы из младшего разряда соседней слева группы в старший разряд данной 4-разрядной группы. При этом, если, например, в процессе сдвига передается единица из R_2'' в R_2', то ее вес в R_2'' был равен 10, а в результате сдвига его значение должно быть уменьшено в два раза и, следовательно, должно быть равно 5. А так как единица, поступающая в старший разряд группы R_2', будет иметь вес 8, то потребуется коррекция вычитанием возникающего избытка в три единицы.

Если число разрядов в регистре R_1 равно n, то преобразование завершается после выполнения серии из n сдвигов.

В таблице 1.2 показан процесс преобразования десятичного числа 235_{10} в двоичную систему счисления.

Таблица 1.2

R_2'''	R_2''	R_2'	R_1	
0 0 1 0	0 0 1 1	0 1 0 1		Исходное состояние
0 0 0 1	0 0 0 1	_1 0 1 0	1	Сдвиг
		0 0 1 1		Коррекция
		0 1 1 1		
0 0 0 0	_1 0 0 0	_1 0 1 1	1 1	Сдвиг
	0 0 1 1	0 0 1 1		Коррекция
	0 1 0 1	1 0 0 0		
0 0 0 0	0 0 1 0	_1 1 0 0	0 1 1	Сдвиг
		0 0 1 1		Коррекция
		1 0 0 1		
0 0 0 0	0 0 0 1	0 1 0 0	1 0 1 1	Сдвиг
0 0 0 0	0 0 0 0	_1 0 1 0	0 1 0 1 1	Сдвиг
		0 0 1 1		Коррекция
		0 1 1 1		
0 0 0 0	0 0 0 0	0 0 1 1	1 0 1 0 1 1	Сдвиг
0 0 0 0	0 0 0 0	0 0 0 1	1 1 0 1 0 1 1	Сдвиг
0 0 0 0	0 0 0 0	0 0 0 0	1 1 1 0 1 0 1 1	Сдвиг

На рис. 1.2 и 1.3 показаны алгоритмы преобразования чисел рассмотренным способом из двоичной системы счисления в десятичную и обратно.

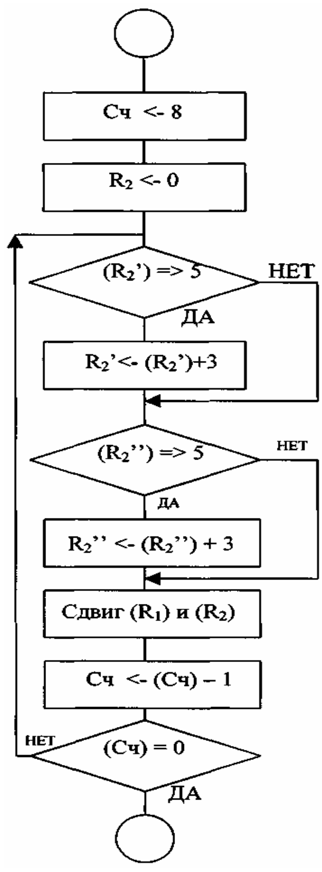


Рис. 1.2

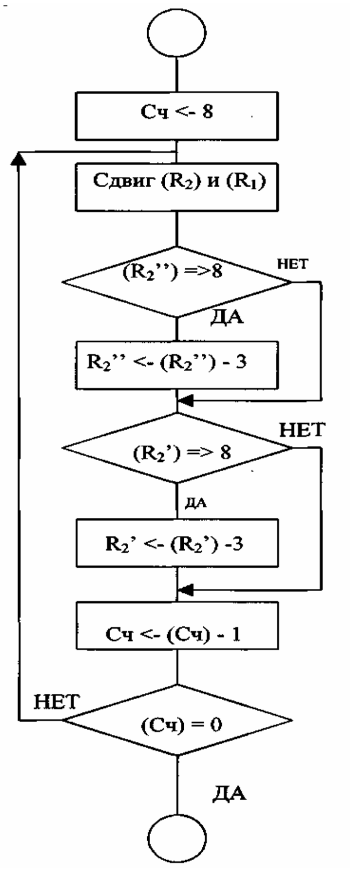


Рис. 1.3

1.3. Упражнения и задания

Таблица 1.3

Номер варианта	A	B	C	D
1	$5170{,}236_8$	$A39{,}FB4_{16}$	$4037{,}587_{10}$	$1001111100{,}110111_2$
2	$6304{,}352_8$	$FBA{,}975_{16}$	$3987{,}654_{10}$	$1011111000{,}101110_2$
3	$2736{,}503_8$	$EFO{,}B94_{16}$	$3095{,}743_{10}$	$1000111101{,}110111_2$
4	1651_8	$1D7{,}\ 6C_{16}$	260_{10}	10111101_2

Над числами A-D, представленными в таблице 1.3, провести следующие операции:

- преобразование восьмеричных чисел в двоичную систему счисления;
- преобразование шестнадцатеричных чисел в двоичную систему счисления;
- преобразование двоичных чисел в восьмеричную систему счисления;
- преобразование двоичных чисел в шестнадцатеричную систему счисления;
- преобразование восьмеричных чисел в десятичную систему счисления;
- преобразование шестнадцатеричных чисел в десятичную систему счисления;
- преобразование двоичных чисел в десятичную систему счисления;
- преобразование десятичных чисел в шестнадцатеричную систему счисления.

2. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ В ЦИФРОВЫХ УСТРОЙСТВАХ

2.1. Целые числа

Числа в цифровых устройствах могут представляться в форме целых чисел, чисел с фиксированной запятой и чисел с плавающей запятой.

При решении задач целые числа встречаются в случаях представления индексов переменных, подсчета числа повторений каких-либо действий и т. д. Для хранения целых чисел в ячейке памяти предусматривается распределение разрядов (разрядная сетка), показанное на рис. 2.1, *а.* Один из *n* разрядов отводится под знак числа, остальные разряды отводятся под модуль числа.

Обычно применяют следующий способ кодирования знака числа: "+" обозначают цифрой 0 в знаковом разряде, "—" – цифрой 1 в знаковом разряде.

Модуль числа занимает в разрядной сетке ее младшие разряды, свободные старшие разряды заполняются нулями. Например, число -13_{10}, представленное в двоичной системе счисления значением -1101_2, в 8-разрядной сетке имеет вид, показанный на рис. 2.1, *б.*

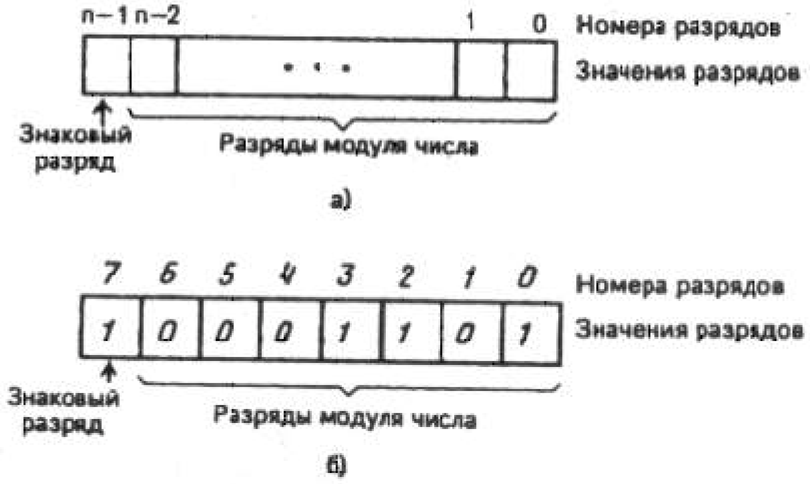


Рис. 2.1

Если количество значащих разрядов модуля числа превышает n–1, происходит потеря старших разрядов модуля. Это явление, называемое *переполнением разрядной сетки,* приводит к ошибке в представлении числа.

Диапазон модулей чисел, которые могут быть представлены в n-разрядной сетке, от 0 (при цифре 0 во всех разрядах модуля) до $2^{n-1} - 1$ (при цифре 1 во всех разрядах модуля).

В универсальных ЭВМ обычно используется два формата целых чисел: *короткий с* числом разрядов n=16 и длинный с n=32. При этом максимальные значения модулей чисел соответственно

$$2^{15} - 1 = 2^5 \cdot 2^{10} - 1 = 32 \cdot 1024 - 1 \approx 32 \cdot 10^3 \text{ (при } n = 16\text{)},$$
$$2^{31} - 1 = 2 \cdot 2^{30} - 1 \approx 2 \cdot 10^{30 \cdot 0,3} = 2 \cdot 10^9 \quad \text{(при } n = 32\text{)}.$$

2.2. Числа с фиксированной точкой

При этой форме обычно запятая, отделяющая целую часть числа от ее дробной части, фиксируется перед старшим разрядом модуля числа (рис. 2.2). Таким образом, значение модуля числа всегда оказывается меньше единицы. Это условие путем выбора определенных *масштабных коэффициентов* должно выполняться для исходных данных задачи и всех промежуточных результатов вычислений.

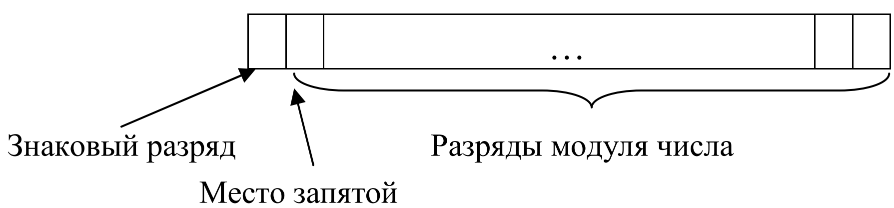


Рис. 2.2

При занесении числа в ячейку памяти свободные младшие разряды заполняются нулями, а если число значащих разрядов модуля больше $n-1$, то младшие разряды модуля, которые не поместились в разрядной сетке, теряются. Это приводит к погрешности, значение которой меньше единицы младшего разряда разрядной сетки, т. е. $\varepsilon_{абс} < 2^{-(n-1)}$.

При n=16 $\varepsilon_{абс} < 2^{-15} = 1/(32\cdot10^3)$, при n=32 $\varepsilon_{абс} < 2^{-31} = 1/(2\cdot10^9)$.

Если число имеет целую часть, то для ее хранения в разрядной сетке места нет, она теряется, число в разрядной сетке оказывается ошибочным.

Достоинство представления чисел в форме с фиксированной запятой состоит в простоте выполнения арифметических операций; недостатки – в необходимости выбора масштабных коэффициентов и в низкой точности представления чисел с малыми значениями модуля (нули в старших разрядах модуля приводят к уменьшению количества разрядов, занимаемых значащей частью модуля числа).

2.3. Числа с плавающей точкой

Для научно-технических расчетов необходимо представлять числа в широком диапазоне и с достаточно большой точностью. Указанным требованиям отвечают числа с плавающей точкой (рис. 2.3).

Рис. 2.3

Число состоит из *мантиссы,* старший разряд которой определяет знак числа, и *порядка* со знаком. Значение модуля мантиссы представляется двоичным дробным числом, т. е. запятая фиксируется перед старшим разрядом модуля мантиссы, порядок представляется целым числом. Порядок указывает действительное положение запятой в числе. Код в приведенном формате представляет значение числа в полулогарифмической форме:

$$N = M \cdot 2^{П},$$

где M и $П$ – мантисса и порядок числа.

Точность представления значений зависит от количества значащих цифр мантиссы. Для повышения точности числа с плавающей запятой представляются в *нормализованной* форме, при которой значение модуля мантиссы лежит в пределах $0{,}5 \le |M| < 1$. Признаком нормализованного числа служит наличие единицы в старшем разряде модуля мантиссы. В нормализованной форме могут быть представлены все числа из некоторого диапазона за исключением нуля.

Нормализованные двоичные числа с плавающей запятой представляют значения модуля чисел в диапазоне

$$0{,}5 \cdot 2^{-\Pi_{\max}} \le |N| \le \left(1 - 2^{m-1}\right) 2^{\Pi_{\max}} \approx 2^{\Pi_{\max}}$$

где $\Pi_{\max} = 2^{p-1}$ – максимальное значение модуля порядка.

Так, при p=7 $\Pi_{\max} = 2^{p-1} - 1 = 2^6 - 1 = 63$ и диапазон представления модулей нормализованных чисел

$$|N|_{\min} = 0{,}5 \cdot 2^{-63} = 2^{-64} \approx 10^{-64 \cdot 0{,}3} \approx 10^{-19},$$

$$|N|_{\max} = 2^{-63} \approx 10^{-63 \cdot 0{,}3} \approx 10^{19}.$$

Таким образом, диапазон чисел от 10^{-19} до 10^{19}.

Для расширения диапазона представляемых чисел при фиксированной длине разрядной сетки (m+p) в качестве основания системы счисления выбирается 2^4=16. При этом число, представляемое в разрядной сетке, приобретает значение N=M·16^{Π}. Нормализованная мантисса шестнадцатеричного числа с плавающей запятой имеет значение, лежащее в диапазоне $1/16 \le |M| < 1$. Признаком нормализации такого числа является наличие хотя бы одной единицы в четырех старших разрядах модуля мантиссы. Диапазон представления чисел в этом случае существенно расширяется, находясь при том же количестве разрядов порядка в пределах от 10^{-75} до 10^{75}.

Рассмотрим погрешность представления чисел с плавающей точкой. Абсолютная погрешность представления числа

$$\varepsilon_{абс} = 2^{-(m-1)} \cdot 2^{p}.$$

Предельная относительная погрешность – отношение абсолютной погрешности к числу при минимальном значении модуля мантиссы нормализованного числа:

$$\varepsilon_{отн} = \frac{\varepsilon_{абс}}{|M|_{\min} \cdot 2^{p}} = \frac{2^{-(m-1)} \cdot 2^{p}}{0{,}5 \cdot 2^{p}} = 4 \cdot 2^{m}.$$

Отсюда видно, что точность представления чисел определяется количеством разрядов, отводимых в разрядной сетке под мантиссу.

В современных ЭВМ числа с плавающей запятой имеют основание системы счисления 16 и представляются в двух форматах: *коротком* (с числом разрядов 32) и *длинном* (с числом разрядов 64). Длинный формат предусматривает увеличение количества разрядов, отводимых в разрядной сетке под мантиссу, за счет чего повышается точность представления чисел.

2.4. Десятичные числа

Для кодирования десятичных чисел используются слова переменной длины с применением двух видов формата: *упакованного* и *распакованного.*

Каждая десятичная цифра представляется двоичной тетрадой и занимает в разрядной сетке четыре разряда. Четыре разряда отводятся и для представления знака (собственно знак представляется младшим разрядом тетрады, в остальных разрядах тетрады может использоваться постоянная комбинация 110).

При использовании упакованного формата каждый *байт* (8 разрядов двоичного числа) содержит две десятичные цифры (рис. 2.4, *а*).

7…4	3…0		7…4	3…0	7…4	3…0
Цифра	Цифра	…	Цифра	Цифра	Цифра	Цифра
байт			байт		байт	

а)

7…4	3…0		7…4	3…0	7…4	3…0
Зона	Цифра	…	Зона	Цифра	Знак	Цифра
байт			байт		байт	

б)

Рис. 2.4

Например, число -6378_{10} представляется в упакованном формате в следующем виде:

0000 0110 0011 0111 1000 1101.
байт *байт* *байт*

В распакованном формате каждый байт содержит лишь одну десятичную цифру в младшей тетраде; старшая тетрада, называемая *зоной*, заполняется стандартной комбинацией 1111 (рис. 2.4, *б*). Число -6378_{10} представляется в этом формате в следующем виде:

1111 0110 1111 0011 1111 0111 1101 10000.
байт *байт* *байт* *байт*

3. ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

3.1. Машинные коды чисел

Главным преимуществом двоичной арифметики является простота выполнения арифметических операций. В частности, при двоичной системе счисления операция вычитания легко заменяется сложением. Для этого используются некоторые специальные коды чисел, суть которых поясним на простых примерах.

Пусть суммируются числа x и y, например, меньшие 100, представленные в десятичном алфавите. Примем $x = 93$, $y = -24$, тогда $x+y = 93-24 = 69$. Поступим теперь иначе. Заменим –24 суммой –24+100. Тогда x+y = 93+(–24+100) = 169. Чтобы получить нужный результат, следует это число уменьшить на 100, что равносильно отбрасыванию полученной в старшем разряде единицы. Такой способ вычитания называется сложением уменьшаемого с *точным дополнением* вычитаемого и последующим отбрасыванием единицы в старшем разряде. Точное дополнение при этом получается дополнением цифр всех разрядов, кроме самого младшего, до основания $h - 1$, а младшего – до h.

Другой вид дополнения, называемый *поразрядным*, получается заменой каждой цифры числа ее дополнением до старшей цифры системы счисления. Для предыдущего примера поразрядное дополнение числа 24 есть число 75 (75 = –24+99). Тогда 93+75=168. Результат имеет лишнюю единицу в старшем разряде и нехватку единицы в младшем. Для получения правильного ответа единицу старшего разряда необходимо изъять и добавить ее к младшему разряду. Эта процедура называется *циклическим переносом*. Истинный результат в рассматриваемом примере получается следующим образом:

```
  93                    93
  76                    75
 169                   168
  ↓                    └──> 1  единица переносится из старшего
                           69  разряда в младший
лишняя единица
отбрасывается
```

То же имеет место и в двоичной арифметике ЭВМ, где для представления двоичных чисел применяют прямой, дополнительный (точной дополнение) и обратный (поразрядное дополнение) коды. Для обозначения знака чисел используется специальный разряд, цифра которого равна нулю, если число положительно, и единице, если число отрицательно.

В дальнейшем для простоты изложения будем оперировать целыми (положительными и отрицательными) числами.

Прямой код. Целое число x в прямом коде $[x]_{\text{пр}}$ представляется в виде

$$[x]_{\text{пр}} = 0\varepsilon_1\varepsilon_2...\varepsilon_n, \text{ если } x \geq 0,$$

$$[x]_{\text{пр}} = 1\varepsilon_1\varepsilon_2...\varepsilon_n, \text{ если } x\langle 0,$$

где ε – двоичная цифра 0 или 1.

Пример 3.1.

Пусть x = +10110, тогда $[x]_{\text{пр}}$= 010110; для x = –11011, $[x]_{\text{пр}}$= 111011.

Обратный код. Если двоичное число $x=\varepsilon_1\varepsilon_2\ldots\varepsilon_n$ является положительным (x>0), то обратный код этого числа совпадает с прямым.

Если x<0, то обратный код получают следующим образом. В знаковом разряде записывается единица, а все остальные разряды заменяются дополнениями до единицы. Следовательно,

$$[x]_{обр} = 1\varepsilon_1^*\varepsilon_2^*...\varepsilon_n^*,$$

где $\varepsilon_1^* = 1$, если ε_i=0, и $\varepsilon_i^* = 0$, если ε_1=1.

Таким образом, для отрицательных чисел обратный код получается заменой нулей единицами и обратно и дополнением единицы в знаковом разряде.

Пример 3.2.

Пусть $x = -101$, тогда $[x]_{обр}$=1010.

Дополнительный код. Дополнительный код числа $[x]_{доп}$ совпадает с самим числом, если $x \geq 0$. Если $x<0$, то дополнительный код получают по следующему правилу: находится обратный код числа и к последнему младшему разряду прибавляется единица.

Пример 3.3.

Пусть $x = -11010$, тогда $[x]_{доп}$ =100101+00001=100110.

3.2. Сложение и вычитание чисел в ЭВМ с использованием кодов

Благодаря введению кодов операция чисел заменяется сложением обратных или дополнительных кодов.

Рассмотрим примеры сложения и вычитания двоичных чисел x и y и их десятичных эквивалентов в различных кодах и с помощью этих примеров установим соответствующие правила. Предположим, что технические возможности ЭВМ не позволяют оперировать числами, большими 63 (т. е. n = 6). Если окажется, что $x+y > 63$, то такое событие называют *переполнением* разрядной сетки.

Пример 3.4.

Даны $x = 22$, $y = 34$, $x+y = 56$.

$$
\begin{array}{r}
[x]_{доп} = 0\,0\,1\,0\,1\,1\,0 = [x]_{обр} \\
+ \\
[y]_{доп} = \underline{0\,1\,0\,0\,0\,1\,0} = [y]_{обр} \\
\underline{0}\ \underline{1\,1\,1\,0\,0\,0}
\end{array}
$$

$$\text{знак} \quad 1\cdot2^5+1\cdot2^4+1\cdot2^3+0\cdot2^2+0\cdot2^1+0\cdot2^0=56$$

Так как числа положительные, то обратный и дополнительный коды одинаковы, знаковый разряд равен нулю.

Отрицательное число y заменяем обратным или дополнительным кодом.

Пример 3.5.

Дано $x = 54$, $y = -23$, $x+y = 31$.

$[x]_{обр}$=0 1 1 0 1 1 0
+
$[y]_{обр}$=1 1 0 1 0 0 0
1 0 0 1 1 1 1 0
+ 1
0 0 1 1 1 1 1

$[x]_{доп}$=0 1 1 0 1 1 0
+
$[y]_{доп}$=1 1 0 1 0 0 1
1 0 0 1 1 1 1 1

В обратном коде единица переполнения разрядной сетки прибавляется к младшему разряду, в дополнительном – отбрасывается.

Пример 3.6.

Дано $x = 23$, $y = –54$, $x+y = –31$. Суммирование двух чисел, одно из которых отрицательно и больше по модулю чем положительное.

$[x]_{обр}$=0 0 1 0 1 1 1
+
$[y]_{обр}$=1 0 0 1 0 0 1
1 1 0 0 0 0 0 0
Минус

$[x]_{доп}$=0 0 1 0 1 1 1
+
$[y]_{доп}$=1 0 0 1 0 1 0
1 1 0 0 0 0 1
Минус

Преобразование в прямой код 0 0 0 0 0 1
1 0 0 0 0 0

-(0 1 1 1 1 1)$_2$=(31)$_{10}$

Пример 3.7.

Дано $x = –23$, $y = –31$, $x+y = –54$, (суммирование отрицательных чисел $x < 0$, $y < 0$, $x+y < 0$). Замена слагаемых их обратными или дополнительными кодами.

Обратные коды	**Дополнительные коды**
$[x]_{обр}$ = **1 1 0 1 0 0 0**	$[x]_{доп}$ = **1 1 0 1 0 0 1**
$[y]_{обр}$ = 1 1 0 0 0 0 0	$[y]_{доп}$ = 1 1 0 0 0 0 1
$[x+y]_{обр}$ = 1 1 0 0 1 0 0 0	$[x+y]_{доп}$ = 1 1 0 0 1 0 1 0
минус + 1	минус
1 0 0 1 0 0 1	0 0 1 0 0 1

-(1 1 0 1 1 0)$_2$= -(54)$_{10}$= -(1 1 0 1 1 0)$_2$

В примерах 3.6 и 3.7 при переходе от дополнительного кода к прямому использовалось вычитание единицы и замена всех нулей единицами и наоборот. Часто такой переход осуществляется образованием дополнительного кода от дополнительного. Можно легко проверить, что результат при этом не искажается.

Если сумма превышает установленный модуль числа (здесь 63), то возникает переполнение разрядной сетки. При этом результат оказывается искаженным.

Пример 3.8.

Дано $x = 46$, $y = 32$, $x+y = 78 > 63$.

$$
\begin{array}{r}
[x]_{обр} = 0\,1\,0\,1\,1\,1\,0 \\
[y]_{обр} = \underline{0\,1\,0\,0\,0\,0\,0} \\
[x]_{обр}+[y]_{обр} = 1\,0\,0\,1\,1\,1\,0
\end{array}
$$

знак

Признаком переполнения разрядной сетки для положительных чисел является возникновение единицы переноса в знаковый разряд при отсутствии переноса из знакового разряда.

Пример 3.9.

Дано x = –46, y = –32, x+y = –78.

Обратный код

$$
\begin{array}{r}
x_{обр} = 1\,0\,1\,0\,0\,0\,1 \\
+ \\
y_{обр} = \underline{1\,0\,1\,1\,1\,1\,1} \\
[x]_{обр}+[y]_{обр} = 1\,0\,1\,1\,0\,0\,0\,0
\end{array}
$$

Знак

Признаком переполнения разрядной сетки для отрицательных чисел является наличие единицы переноса из знакового разряда при отсутствии единицы переноса в знаковый ряд.

Эти признаки переполнения используются в ЭВМ для обнаружения результатов алгебраического сложения, не представимых из-за конечного числа разрядов в разрядной сетке.

Подведем итоги рассмотренных примеров. При *сложении* образуют коды слагаемых – обратные или дополнительные – по известным правилам и складывают их. При этом выполняется циклическое сложение, если используются обратные коды, или отбрасывание старшей единицы переноса – при сложении в дополнительных кодах. Если переполнения нет, т. е. указанные в примерах 3.8 и 3.9 переносы одинаковы, полученная сумма кодов есть код – обратный или дополнительный – искомой суммы. Он может быть переведен в прямой.

При *вычитании* знак вычитаемого изменяется на противоположный, а далее выполняются описанные выше действия.

При переполнении результат как при сложении, так и при вычитании искажается. В ЭВМ предусматривается определенная реакция на это событие – прерывание.

Чаще всего в ЭВМ используются дополнительные коды. В них нуль представляется только одним способом (при обратных кодах «положительный» и «отрицательный» нуль изображаются различно). Кроме того, при дополнительных кодах нет циклического сложения, что ускоряет операцию.

3.3. Сложение чисел, представленных в форме с плавающей точкой

Предварительно отметим, что при выполнении всех операций над числами с плавающей точкой результат должен быть нормализован. При сложении возможны различные варианты результата. Проследим последовательность выполнения сложения над числами на простом примере.

Первой операцией, которая производится при сложении двух чисел, является операция выравнивания порядков. Одинаковые порядки слагаемых позволяют осуществить суммирование только одних мантисс. При выравнивании порядок числа изменяется таким образом, что при каждом сдвиге мантиссы на один разряд вправо порядок увеличивается на единицу. Понятно, что для обеспечения большей точности выравнивается число с меньшим порядком, но нормализация у него при этом нарушается. Затем складываются мантиссы и результат при необходимости нормализуется. Допустим, что даны два слагаемых, записанных в следующей форме:

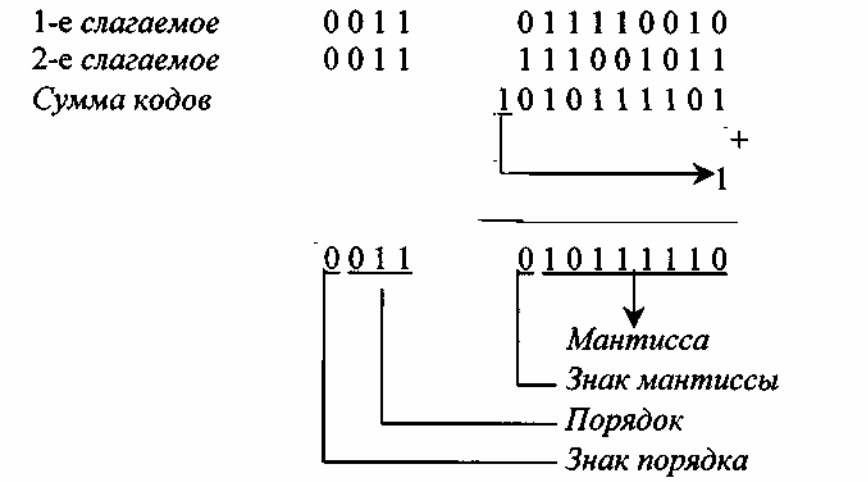
	Знак	Порядок	Знак мантиссы	Мантисса
1-е слагаемое	0	011	0	11110010
2-е слагаемое	0	001	1	11010011

Шаг 1. Выравнивание порядка второго слагаемого (денормализация) с помощью сдвига его мантиссы вправо на два разряда, подсуммирование двух единиц к разрядам порядка. Тогда второе слагаемое запишется в виде:

0 011 1 00110100.

Отметим, что в результате денормализации младшие разряды оказались потерянными.

Шаг 2. Перевод мантисс в один из кодов (например, обратный) и их сложение.



Полученное в результате сложений число является положительным и нормализованным (в первом значащем разряде мантиссы стоит единица). В таком виде оно может быть использовано для дальнейших вычислений.

3.4. Умножение чисел в ЭВМ

При умножении двух n-разрядных чисел, представленных в форме с фиксированной точкой, получается $2n$-разрядное произведение. Так как длина разрядной сетки машины ограничена конечным числом разрядов, то полученный результат всегда будет иметь некоторую погрешность. Кроме того, необходимо помнить, что при умножении целых чисел размерность сомножителей должна обеспечивать размерность результата, не превышающего разрядную сетку ЭВМ. В противном случае возникает переполнение. Процесс умножения, как известно, может быть представлен сложением со сдвигом. Если сдвиг осуществлять в правую сторону, то младшие разряды произведения будут теряться и произведение, как правило, будет получаться с недостатком. При многократном выполнении умножения систематическая ошибка будет накапливаться, что исказит окончательный результат.

Одним из приемов уменьшения этой ошибки является округление результата. Оно заключается в том, что после умножения к произведению подсуммируется единица, если дополнительный $(n+1)$-й разряд равен единице, и ничего не добавляется, если он равен нулю. Таким образом, округление производится попеременно в большую или меньшую сторону.

При умножении знак произведения можно определить как результат логической операции неравнозначности над знаковыми разрядами сомножителей.

Пусть множимое M_1 и множитель M_2 представлены многочленами

$$M_1 = a_1 \cdot 2^{-1} + a_2 \cdot 2^{-2} + ... + a_n \cdot 2^{-n},$$
$$M_2 = b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + ... + b_n \cdot 2^{-n}.$$

В представленных многочленах a_i и b_i – коэффициенты, равны нулю и единице. Произведение сомножителей можно записать так:

$$M_1 \cdot M_2 = M_1 \cdot 2^{-1} \cdot b_1 + M_1 \cdot 2^{-2} \cdot b_2 + \ldots + M_1 \cdot 2^{-n} \cdot b_n.$$

Таким образом, сдвигая последовательно множимое M_1 на один разряд вправо (т. е. уменьшая его каждый раз вдвое) и суммируя частичные произведения, получаем величину $M_1 \cdot M_2$. Заметим, что при $b_i = 0$ частичным произведением будет нуль.

Пример 3.10.

Перемножить с фиксированной запятой два числа $M_1 = 0{,}10011$ и $M_2=0{,}11001$.

0 10011	M_1
×	
0 11001	$M_2=b_1 \cdot 2^{-1}+b_2 \cdot 2^{-2}+b_3 \cdot 2^{-3}+b_4 \cdot 2^{-4}-b_5 \cdot 2^{-5}$
0 01001	$M_1 \cdot 2^{-1} \cdot b_1$
+	
0 0010011	$M_1 \cdot 2^{-2} \cdot b_2$
0 0111001	$M_1 \cdot 2^{-1} \cdot b_1 + M_1 \cdot 2^{-2} \cdot b_2$
+	
0 0000000	$M_1 \cdot 2^{-3} \cdot b_3$ $(b_3=0)$
0 0111001	$M_1 \cdot 2^{-1} \cdot b_1 + M_1 \cdot 2^{-2} \cdot b_2 + M_1 \cdot 2^{-3} \cdot b_3$
+	
0 0000000	$M_1 \cdot 2^{-4} \cdot b_4$ $(b_4=0)$
0 0111001	$M_1 \cdot 2^{-1} \cdot b_1 + M_1 \cdot 2^{-2} \cdot b_2 + M_1 \cdot 2^{-3} \cdot b_3 + M_1 \cdot 2^{-4} \cdot b_4$
+	
0 000001001	$M_1 \cdot 2^{-5} \cdot b_5$
0 011101101	$M_1 \cdot 2^{-1} \cdot b_1 + M_1 \cdot 2^{-2} \cdot b_2 + M_1 \cdot 2^{-3} \cdot b_3 + M_1 \cdot 2^{-4} \cdot b_4 + M_1 \cdot 2^{-5} \cdot b_5 = M_1 \cdot M_2$

Умножение чисел с плавающей точкой осуществляется в четыре этапа:

- определение знака произведения;
- определение порядка произведения (алгебраическое сложение порядков сомножителей);
- перемножение мантисс;
- нормализация.

При выполнении операции следят за переполнением порядка.

3.5. Деление чисел в ЭВМ

Деление в ЭВМ сводится к последовательности вычитаний делителя сначала из делимого, а затем из образующихся в процессе деления частичных остатков и их сдвига.

При делении чисел с фиксированной точкой сначала устанавливают, не является ли делимое x большим делителя y. Если же переполнения быть не может, процесс деления описывается следующим образом.

При $x–y < 0$ первая цифра частного, т.е. его целая часть, равна нулю. Чтобы получить первую цифру после запятой, необходимо из делимого вычесть делитель, сдвинутый на один разряд вправо. Если получена положительная разность, то в разряд частного записывается единица, а из полученного остатка вновь вычитается сдвинутый делитель. При отрицательном остатке необходимо в частное записать нуль, вновь восстановить предыдущий остаток и вычесть из него сдвинутый еще на один разряд делитель и т.д. Процесс деления может быть закончен, когда получено число цифр после запятой, не меньшее чем у заданных делимого и делителя.

Чтобы восстановить предыдущий (k–1)-й остаток, достаточно сложить k-й остаток с делителем, сдвинутым на k разрядов вправо, т.е.

$$x_{k-1} = x_k + y \cdot 2^{-k}.$$

Следующий (k+l)-й остаток можно получить вычитанием из восстановленного (k–1)-го остатка делителя сдвинутого еще на один разряд вправо:

$$x_{k+1} = x_{k-1} - y \cdot 2^{-(k+1)}.$$

Описанный способ называется *делением с восстановлением остатка*, так как при получении отрицательного остатка для продолжения деления действительно приходится возвращаться к предыдущему остатку.

Пример 3.11.

Разделить x = +0,101001 на y+0,110100.

Делимое меньше делителя, первая цифра частного (целая часть) 0	0,101001 | 0,110100 0,11001
Сдвиг делителя вправо на один разряд	0,0110100
1-й остаток, равный разности делимого и сдвинутого делителя, положителен, следовательно, первая цифра частного – единица	+0,0011110
Сдвиг делителя вправо на один разряд	0,00110100
2-й остаток положителен, в разряде частного единица	+0,00001000
Сдвиг делителя вправо на один разряд	0,00011010
3-й остаток отрицателен, в разряд частного записывается нуль	–0,00010010 +
Восстановление второго остатка	0,00011010 +0,00001000
Сдвиг делителя вправо на один разряд	0,00001101
4-й остаток отрицателен, в разряде частного нуль	–0,00000101 +
Повторное восстановление предыдущего остатка	0,00000101 0,00001000
Сдвиг делителя вправо на один разряд	0,000001101
5-й остаток положителен, в разряд частного записывается единица	0,000000011

На этом шаге процесс может быть остановлен, так как ясно, что следующая цифра частного будет равна нулю, а число полученных значащих разрядов равно числу разрядов делимого. Аналогично осуществляется процесс деления чисел, содержащих целую часть.

Для получения любого промежуточного (k+1)-го остатка можно обойтись без восстановления предыдущего. Для этого достаточно к остатку x_k прибавить делитель, сдвинутый еще на один разряд вправо, т.е.

$$x_{k+1} = x_k + y \cdot 2^{-(k+1)}$$

или

$$x_{k+1} = x_{k-1} - y \cdot 2^{-k} + 2^{-(k+1)} = x_{k-1} - y \cdot 2^{-(k+1)} \cdot \left(2^1 - 1\right).$$

Так как выражение, стоящее в скобках – единица, то $x_{k+1} = x_{k-1} - y \cdot 2^{-(k+1)}$. Такой способ называется *делением без восстановления остатка*. Благодаря более быстрой реализации он наиболее часто используется в ЭВМ.

Знак частного находится так же, как и в операции умножения.

Деление чисел с плавающей точкой выполняется следующим образом:

- определяется знак частного;
- вычисляется порядок частного путем вычитания из порядка делимого порядка делителя, при этом следят за переполнением;
- определяется мантисса частного делением мантисс;
- осуществляется нормализация результата.

3.6. Упражнения и задания

Выполнить арифметические операции, используя в качестве операндов числа A и B, представленные в таблице 3.1.

Таблица 3.1

Номер варианта	*A*	*B*
1	+0,0101111	+0,1100110
2	-0,0111101	+0,1101101
3	+0,0100101	-0,1010111
4	-0,0110011	-0,1001101

- Найти сумму *A*+*B*.
- Найти разность *A*–*B*.
- Вычислить произведение *A*·*B*, используя для представления мантисс операндов: *а*) прямой код; *б*) дополнительный код.
- Вычислить частное *A*/*B*.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Грицевский П. М. Основы автоматики, импульсной и вычислительной техники: Учебник для техникумов/ П. М. Грицевский, А. Е. Мамченко, Б. М. Степенский. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1987. – 384 с.
2. Калабеков Б. А. Цифровые устройства и микропроцессорные системы: Учебник для техникумов связи/ Б. А. Калабеков. – М.: Радио и связь, 1997. – 336 с.

Учебное издание

Арифметические основы цифровой техники

Методические указания

Составители: ГОРБОКОНЕНКО Вера Дмитриевна, ШИКИНА Виктория Евгеньевна

Редактор Н. А. Евдокимова

Подписано в печать 01.10.2003. Формат 60x84/16.
Бумага газетная. Печать трафаретная. Усл. печ. л. 1,63.
Уч.-изд. л. 1,50. Тираж 100 экз. Заказ

Ульяновский государственный технический университет
432027, г. Ульяновск, у. Сев. Венец, д. 32
Типография УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.