

Кафедра "Информатика"

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к практическим занятиям по курсу "Информатика"
по теме:
"Арифметические основы ЭВМ "**

**Ростов-на-Дону
2004**

Составитель: Красников В.В.

Методические указания к практическим занятиям по курсу "Информатика" по теме "Арифметические основы ЭВМ". Ростов н/Д: ДГТУ, 2004 – 22 с.

Методические указания предназначены для проведения практических занятий по курсу "Информатика" у студентов специальностей 071900 и 351400, а также для студентов других специальностей, в той или иной степени знакомящихся с организацией ЭВМ. В методических указаниях рассмотрены принципы построения позиционных систем счисления, правила перевода чисел из одной системы счисления в другую, двоичная арифметика, а также основные принципы хранения и обработки числовой информации в ЭВМ. Методические указания содержат большое число примеров и заданий.

Печатается по решению методической комиссии факультета "Автоматизация и информатика"

Рецензент: к.ф.-м.н. Бычков А.А.

© ДГТУ

1. Системы счисления.

1.1 Основные понятия и определения.

Под **системой счисления** понимается способ представления любого числа с помощью некоторого алфавита символов, называемых цифрами.

Все системы счисления делятся на позиционные и непозиционные.

Непозиционными системами являются такие системы счисления, в которых каждый символ сохраняет свое значение независимо от места его положения в числе.

Примером непозиционной системы счисления является римская система. К недостаткам таких систем относятся наличие большого количества знаков и сложность выполнения арифметических операций.

Система счисления называется **позиционной**, если одна и та же цифра имеет различное значение, определяющееся позицией цифры в последовательности цифр, изображающей число. Это значение меняется в однозначной зависимости от позиции, занимаемой цифрой, по некоторому закону.

Примером позиционной системы счисления является десятичная система, используемая в повседневной жизни.

Количество p различных цифр, употребляемых в позиционной системе определяет название системы счисления и называется **основанием** системы счисления - " p ".

В десятичной системе используются десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; эта система имеет основанием число десять.

Любое число N в позиционной системе счисления с основанием p может быть представлено в виде полинома от основания p :

$$N = a_k p^k + a_{k-1} p^{k-1} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + a_{-2} p^{-2} + \dots \quad (1.1)$$

здесь N - число, a - коэффициенты (цифры числа), p - основание системы счисления ($p > 1$).

Принято представлять числа в виде последовательности цифр:

$$N = a_k a_{k-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots$$

В этой последовательности точка отделяет целую часть числа от дробной (коэффициенты при положительных степенях, включая нуль, от коэффициентов при отрицательных степенях). Точка опускается, если нет отрицательных степеней (число целое).

В ЭВМ применяют позиционные системы счисления с недесятичным основанием: двоичную, восьмеричную, шестнадцатеричную.

В аппаратной основе ЭВМ лежат двухпозиционные элементы, которые могут находиться только в двух состояниях; одно из них обозначается 0, а другое - 1.

Поэтому основной системой счисления применяемой в ЭВМ является двоичная система.

Двоичная система счисления. Используется две цифры: 0 и 1. В двоичной системе любое число может быть представлено в виде:

$$X = b_M b_{M-1} \dots b_1 b_0 \cdot b_{-1} b_{-2} \dots ,$$

где b_j либо 0, либо 1.

Эта запись соответствует сумме степеней числа 2, взятых с указанными коэффициентами:

$$X = b_M \cdot 2^M + b_{M-1} \cdot 2^{M-1} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0 + b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + \dots$$

Восьмеричная система счисления. Используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Употребляется в ЭВМ как вспомогательная для записи информации в сокращенном виде. Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада) (Таб. 1).

Шестнадцатеричная система счисления. Для изображения чисел употребляются 16 цифр. Первые десять цифр этой системы обозначаются цифрами от 0 до 9, а старшие шесть цифр – латинскими буквами: 10–А, 11–В, 12–С, 13–D, 14–Е, 15–F. Шестнадцатеричная система используется для записи информации в сокращенном виде. Для представления одной цифры шестнадцатеричной системы счисления используется четыре двоичных разряда (тетрада) (Таб. 1).

Таб. 1. Наиболее важные системы счисления.

Двоичная (Основание 2)	Восьмеричная (Основание 8)	Десятичная (Основание 10)	Шестнадцатеричная (Основание 16)
	Триады		Тетрады
0	0 000	0	0 0000
1	1 001	1	1 0001
	2 010	2	2 0010
	3 011	3	3 0011
	4 100	4	4 0100
	5 101	5	5 0101
	6 110	6	6 0110
	7 111	7	7 0111
		8	8 1000
		9	9 1001
			A 1010
			B 1011
			C 1100
			D 1101
			E 1110
			F 1111

1.2 Перевод чисел из одной системы счисления в другую.

Перевод чисел в десятичную систему осуществляется путем составления степенного ряда с основанием той системы, из которой число переводится. Затем подсчитывается значение суммы.

Пример.

а) Перевести $10101101.101_2 \rightarrow "10" \text{ с.с.}$ *

$$10101101.101_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 173.625_{10}$$

б) Перевести $703.04_8 \rightarrow "10" \text{ с.с.}$

$$703.04_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 + 0 \cdot 8^{-1} + 4 \cdot 8^{-2} = 451.0625_{10}$$

в) Перевести $B2E.4_{16} \rightarrow "10" \text{ с.с.}$

$$B2E.4_{16} = 11 \cdot 16^2 + 2 \cdot 16^1 + 14 \cdot 16^0 + 4 \cdot 16^{-1} = 2862.25_{10}$$

Перевод целых десятичных чисел в восьмеричную, шестнадцатеричную и двоичную системы осуществляется последовательным делением десятичного числа на основание той системы, в которую оно переводится, до тех пор, пока не получится частное меньше этого основания. Число в новой системе записывается в виде остатков деления, начиная с последнего.

Пример.

а) Перевести $181_{10} \rightarrow "8" \text{ с.с.}$

$$\begin{array}{r|l} 181 & 8 \\ \hline 176 & 22 \\ \hline 5 & 16 \\ \hline & 2 \\ \hline & 6 \end{array}$$

Результат $181_{10} = 265_8$.

б) Перевести $622_{10} \rightarrow "16" \text{ с.с.}$

$$\begin{array}{r|l} 622 & 16 \\ \hline 48 & 38 \\ \hline 142 & 32 \\ \hline 128 & 2 \\ \hline & 6 \end{array}$$

↙ 14

Результат $622_{10} = 26E_{16}$.

* Здесь и в дальнейшем при одновременном использовании нескольких различных систем счисления основание системы к которой относится число будем указывать в виде нижнего индекса.

Перевод правильных дробей из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Для перевода правильной десятичной дроби в другую систему эту дробь надо последовательно умножать на основание той системы, в которую она переводится. При этом умножаются только дробные части. Дробь в новой системе записывается в виде целых частей произведений, начиная с первого.

Пример.

Перевести $0.3125_{10} \rightarrow "8" \text{ с.с.}$

$$\begin{array}{r|l} 0 & 3125 \times 8 \\ \hline 2 & 5000 \times 8 \\ 4 & 0000 \end{array}$$

Результат $0.3125_{10} = 0.24_8$.

Замечание. Конечной десятичной дроби в другой системе счисления может соответствовать бесконечная (иногда периодическая) дробь. В этом случае количество знаков в представлении дроби в новой системе берется в зависимости от требуемой точности.

Пример.

Перевести $0.65_{10} \rightarrow "2" \text{ с.с.}$ Точность 6 знаков.

$$\begin{array}{r|l} 0 & 65 \times 2 \\ \hline 1 & 3 \times 2 \\ 0 & 6 \times 2 \\ 1 & 2 \times 2 \\ 0 & 4 \times 2 \\ 0 & 8 \times 2 \\ 1 & 6 \times 2 \\ \dots & \end{array}$$

Результат $0.65_{10} \approx 0.10(1001)_2$.

Для перевода неправильной десятичной дроби в систему счисления с **недесятичным основанием** необходимо отдельно перевести целую часть и отдельно дробную.

Пример. Перевести $23.125_{10} \rightarrow "2" \text{ с.с.}$

1) Переведем целую часть:

$$\begin{array}{r} 2 \mid 2 \\ 3 \mid \\ \hline 2 \mid 1 \mid 2 \\ 2 \mid 1 \mid \\ \hline 1 \mid 5 \mid 2 \\ 1 \mid 0 \mid \\ \hline 1 \mid 4 \mid 2 \mid 2 \\ 1 \mid 1 \mid 2 \mid 1 \\ \hline 0 \end{array}$$

2) Переведем дробную часть:

$$\begin{array}{r|l} 0 & 125 \times 2 \\ \hline 2 & 25 \\ 0 & \times 2 \\ 5 & \\ 0 & \times 2 \\ 0 & \\ 1 & \end{array}$$

Таким образом $23_{10} = 10111_2$; $0.125_{10} = 0.001_2$. Результат: $23.125_{10} = 10111.001_2$.

Необходимо отметить, что целые числа остаются целыми, а правильные дроби – дробями в любой системе счисления.

Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) (Таб. 1) или четырехразрядным двоичным числом (тетрадой) (Таб. 1), при этом отбрасывают ненужные нули в старших и младших разрядах.

Пример.

$$\text{а) } \underbrace{3}_{011} \underbrace{0}_{000} \underbrace{5}_{101} \underbrace{.4}_{1100}_8 = 11000101.1_2;$$

$$\text{б) } \underbrace{7}_{0111} \underbrace{B}_{1011} \underbrace{2}_{0010} \underbrace{.E}_{1110}_{16} = 11110110010.111_2.$$

Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе поступают следующим образом: двигаясь от точки влево и вправо, разбивают двоичное число на группы по три (четыре) разряда, дополняя при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой.

Пример.

а) Перевести $1101111001.1101_2 \rightarrow "8"$ с.с.

$$\underbrace{001}_1 \underbrace{101}_5 \underbrace{111}_7 \underbrace{001}_1 \underbrace{.110}_6 \underbrace{100}_4 = 1571.64_8$$

б) Перевести $1111111011.100111_2 \rightarrow "16"$ с.с.

$$\underbrace{01111111}_7 \underbrace{1011}_F \underbrace{.1001}_B \underbrace{1100}_9 \underbrace{}_C = 7FB.9C_{16}$$

Перевод из восьмеричной в шестнадцатеричную систему и обратно осуществляется через двоичную систему с помощью триад и тетрад.

Пример. Перевести $175.24_8 \rightarrow "16"$ с.с.

$$\underbrace{1}_{001} \underbrace{7}_{111} \underbrace{5}_{101} \underbrace{.2}_{010} \underbrace{4}_{100}_8 = 1111101.0101_2 = \underbrace{01111101}_7 \underbrace{.0101}_D_2$$

$$= 7D.5_{16}$$

Результат: $175.24_8 = 7D.5_{16}$.

1.3 Двоичная арифметика.

Правила выполнения арифметических действий над двоичными числами задаются таблицами двоичных сложения, вычитания и умножения.

Таблица двоичного сложения	Таблица двоичного вычитания	Таблица двоичного умножения
0+0=0	0-0=0	0×0=0
0+1=1	1-0=1	0×1=0
1+0=1	1-1=0	1×0=0
1+1=10	10-1=1	1×1=1

Пример. Выполнить сложение двоичных чисел:

а) X=1101, Y=101;

$$\begin{array}{r}
 11 \\
 X= 1101 \\
 Y=+ \underline{101} \\
 X+Y= 10010
 \end{array}$$

единицы
переноса
←

Результат 1101+101=10010.

При **сложении** двоичных чисел в каждом разряде производится сложение цифр слагаемых и переноса из соседнего младшего разряда, если он имеется. При этом необходимо учитывать, что 1+1 дают нуль в данном разряде и единицу переноса в следующий.

б) X=1101, Y=101, Z=111;

$$\begin{array}{r}
 1 \\
 \swarrow \\
 X= 1101 \\
 Y= + 101 \\
 Z= + \underline{111} \\
 X+Y+Z=11001
 \end{array}$$

единицы
переноса
←

Результат 1101+101+111=11001.

При **вычитании** двоичных чисел в данном разряде при необходимости занимает 1 из старшего разряда. Эта занимаемая 1 равна двум 1 данного разряда.

Пример. Заданы двоичные числа X=10010 и Y=101. Вычислить X-Y.

$$\begin{array}{r}
 10010 \\
 - 101 \\
 \hline
 01101
 \end{array}$$

Результат 10010 - 101=1101.

Умножение двоичных чисел производится по тем же правилам, что и для десятичных с помощью таблиц двоичного умножения и сложения.

Пример. $1001 \times 101 = ?$

$$\begin{array}{r} 1001 \\ \times 101 \\ \hline 1001 \\ 1001 \\ 1001 \\ \hline 101101 \end{array}$$

Результат $1001 \times 101 = 101101$.

Деление двоичных чисел производится по тем же правилам, что и для десятичных. При этом используются таблицы двоичного умножения и вычитания.

Пример.

$1100.011 : 10.01 = ?$

$$\begin{array}{r|l} 110001.1 & 1001 \\ - 1001 & 101.1 \\ \hline 1101 & \\ - 1001 & \\ \hline 1001 & \\ - & \\ \hline 1001 & \\ 0 & \end{array}$$

Результат $1100.011 : 10.01 = 101.1$.

Упражнения 1.

1. Перевести следующие числа в десятичную систему счисления:

а) 110111_2 ; б) 10110111.1011_2 ; в) 563.44_8 ; г) 721.35_8 ; д) $1C4.A_{16}$; е) $9A2F.B5_{16}$.

2. Перевести следующие числа из "10" с.с в "2", "8", "16" с.с.:

а) 463; б) 1209; в) 362; г) 3925; д) 11355.

3. Перевести следующие числа из "10" с.с в "2", "8", "16" с.с. (точность 5 знаков после точки):

а) 0.0625; б) 0.345; в) 0.225; г) 0.725;

д) 217.375; е) 31.2375; ж) 725.03125; з) 8846.04.

4. Перевести следующие числа в двоичную систему счисления:

а) 1725.326_8 ; б) 341.34_8 ; в) $7BF.52A_{16}$; г) $3D2.C_{16}$.

5. Перевести следующие числа из одной системы счисления в другую:

а) $11011001.01011_2 \rightarrow "8" \text{ с.с.};$

б) $1011110.1101_2 \rightarrow "8" \text{ с.с.};$

в) $1101111101.0101101_2 \rightarrow "16" \text{ с.с.};$ г) $110101000.100101_2 \rightarrow "16" \text{ с.с.}.$

6. Перевести следующие числа из одной системы счисления в другую:

а) $312.7_8 \rightarrow "16" \text{ с.с.};$ б) $51.43_8 \rightarrow "16" \text{ с.с.};$

в) $5B.F_{16} \rightarrow "8" \text{ с.с.};$ г) $D4.19_{16} \rightarrow "8" \text{ с.с.}.$

7. Заданы двоичные числа X и Y . Вычислить $X+Y$ и $X-Y$, если:

- а) $X=1101001$, $Y=101111$;
- б) $X=101110110$, $Y=10111001$;
- в) $X=100011001$, $Y=101011$.

8. Заданы двоичные числа X и Y . Вычислить $X*Y$ и X/Y , если:

- а) $X=1000010011$, $Y=1011$;
- б) $X=110010101$, $Y=1001$;
- в) $X=100101.011$, $Y=110.1$;
- г) $X=100000.1101$, $Y=101.01$.

2. Основы машинной арифметики с двоичными числами.

Любая информация (числа, команды, записи и т. п.) представляется в ЭВМ в виде двоичных кодов фиксированной или переменной длины. Отдельные элементы двоичного кода, имеющие значение 0 или 1, называют разрядами или битами. Двоичный код состоящий из 8 разрядов носит название байта. Для записи чисел также используют 32-разрядный формат (машинное слово), 16-разрядный формат (полуслово) и 64-разрядный формат (двойное слово).

2.1 Коды чисел.

В ЭВМ в целях упрощения выполнения арифметических операций применяют специальные коды для представления чисел. Использование кодов позволяет свести операцию вычитания чисел к арифметическому сложению кодов этих чисел. Применяются **прямой**, **обратный** и **дополнительный** коды чисел. Прямой код используется для представления отрицательных чисел в запоминающем устройстве ЭВМ, а также при умножении и делении. Обратный и дополнительный коды используются для замены операции вычитания операцией сложения, что упрощает устройство арифметического блока ЭВМ. К кодам выдвигаются следующие требования:

- 1) Разряды числа в коде жестко связаны с определенной разрядной сеткой.
- 2) Для записи кода знака в разрядной сетке отводится фиксированный, строго определенный разряд. Например, если за основу представления кода взят один байт, то для представления числа будет отведено 7 разрядов, а для записи кода знака один разряд.

Прямой код. Прямой код двоичного числа совпадает по изображению с записью самого числа. Значение знакового разряда для положительных чисел равно 0, а для отрицательных чисел 1.*

Пример. В случае, когда для записи кода выделен один байт, для числа +1101 прямой код 0,0001101, для числа -1101 прямой код 1,0001101.

* Знаковым разрядом обычно является крайний разряд в разрядной сетке. В дальнейшем при записи кода знаковый разряд от цифровых условимся отделять запятой. Если количество разрядов кода не указано будем предполагать, что под запись кода выделен один байт.

Обратный код. Обратный код для положительного числа совпадает с прямым кодом. Для отрицательного числа все цифры числа заменяются на противоположные (1 на 0, 0 на 1), а в знаковый разряд заносится единица.

Пример.

Для числа +1101 прямой код 0, 0001101; обратный код 0, 0001101.

Для числа -1101 прямой код 1, 0001101; обратный код 1, 1110010.

Дополнительный код. Дополнительный код положительного числа совпадает с прямым кодом. Для отрицательного числа дополнительный код образуется путем получения обратного кода и добавлением к младшему разряду единицы.

Пример.

Для числа +1101:

прямой код	обратный код	дополнительный код
0,0001101	0,0001101	0,0001101

Для числа -1101:

прямой код	обратный код	дополнительный код
1,0001101	1,1110010	1,1110011

2.2 Особенности сложения чисел в обратном и дополнительном кодах.

При сложении чисел в дополнительном коде возникающая единица переноса в знаковом разряде отбрасывается.

При сложении чисел в обратном коде возникающая единица переноса в знаковом разряде прибавляется к младшему разряду суммы кодов.

Если результат арифметических действий является кодом отрицательного числа, необходимо преобразовать его в прямой код. При этом обратный код преобразуется в прямой заменой цифр во всех разрядах, кроме знакового, на противоположные. Дополнительный код преобразуется в прямой так же, как и обратный, с последующим прибавлением единицы к младшему разряду.

Пример.

Сложить X и Y в обратном и дополнительном кодах.

а) X= 111, Y= -11;

1) Сложим числа, пользуясь правилами двоичной арифметики:

$$\begin{array}{r} X = 111 \\ Y = -11 \\ \hline X+Y = 100 \end{array}$$

2) Сложим числа, используя коды:

Прямой код	Сложение в обратном	Сложение в дополни-
	коде	тельном коде
X _{пр} =0,0000111	X _{обр} = 0,0000111	X _{доп} = 0,0000111
Y _{пр} =1,0000011	Y _{обр} = 1,1111100	Y _{доп} = 1,1111101
	1 0,0000011	1)0,0000100
	+1	отбрасывается
	(X+Y) _{обр} = 0,0000100	(X+Y) _{доп} = 0,0000100

Так

как

результат сложения является кодом положительного числа (знак 0), то $(X+Y)_{\text{обр}}=(X+Y)_{\text{доп}}=(X+Y)_{\text{пр}}$.

б) $X = -101, Y = -11$;

1) Сложим числа, пользуясь правилами двоичной арифметики:

$$\begin{array}{r} X = -101 \\ Y = -110 \\ \hline X+Y = -1011 \end{array}$$

2) Сложим числа, используя коды:

Прямой код	Сложение в обратном коде	Сложение в дополнительном коде
$X_{\text{пр}} = 1,0000101$	$X_{\text{обр}} = 1,1111010$	$X_{\text{доп}} = 1,1111011$
$Y_{\text{пр}} = 1,0000110$	$Y_{\text{обр}} = 1,1111001$	$Y_{\text{доп}} = 1,1111010$
	$\begin{array}{r} 1\ 1,1110011 \\ \xrightarrow{+1} \\ (X+Y)_{\text{обр}} = 1,1110100 \end{array}$	$\begin{array}{r} 1)1,1110101 \\ \text{отбрасывается} \\ (X+Y)_{\text{доп}} = 1,1110101 \end{array}$

Так как сумма является кодом отрицательного числа (знак 1), то необходимо перевести результаты в прямой код:

а) из обратного кода

$$(X+Y)_{\text{обр}} = 1,1110100 \rightarrow (X+Y)_{\text{пр}} = 1,0001011;$$

б) из дополнительного кода

$$(X+Y)_{\text{доп}} = 1,1110101 \rightarrow (X+Y)_{\text{пр}} = 1,0001010 + 0,0000001 = 1,0001011.$$

Таким образом, $X+Y = -1011$ и полученный результат совпадает с обычной записью

2.3 Модифицированные обратный и дополнительный коды.

При переполнении разрядной сетки, происходит перенос единицы в знаковый разряд. Это приводит к неправильному результату, причем положительное число, получившееся в результате арифметической операции может восприниматься как отрицательное (так как в знаковом разряде "1") и наоборот.

Например:

$$\begin{array}{r} X = 0,1010110 \\ Y = 0,1101000 \\ \hline X+Y = 1,0111110 \end{array}$$

Здесь X и Y – коды положительных чисел, но ЭВМ воспринимает результат их сложения как код отрицательного числа ("1" в знаковом разряде). Для обнаружения переполнения разрядной сетки вводятся модифицированные коды.

Модифицированный обратный код – в нем под знак числа отводится не один, а два разряда. Форма записи чисел в модифицированном обратном коде выглядит следующим образом:

1) для положительного числа

$$X = X_1 X_2 X_3 \dots X_n; \quad X_{\text{обр}}^M = 00, X_1 X_2 X_3 \dots X_n;$$

2) для отрицательного числа

$$X = X_1 X_2 X_3 \dots X_n; \quad X_{об\ p}^M = 11, \bar{X}_1 \bar{X}_2 \bar{X}_3 \dots \bar{X}_n;$$

(обозначение \bar{X} читается “не X”, т.е., если $X=0$, то $\bar{X}=1$ и наоборот, если $X=1$, то $\bar{X}=0$).

В модифицированном обратном и модифицированном дополнительном кодах под знак числа отводится не один, а два разряда: "00" соответствует знаку "+", "11" – знаку "-". Любая другая комбинация ("01" или "10"), получившаяся в знаковых разрядах служит **признаком переполнения разрядной сетки**. Сложение чисел в модифицированных кодах ничем не отличается от сложения в обычных обратном и дополнительном кодах.

Рассмотрим предыдущий пример, выполнив сложение в модифицированном обратном коде:

$$\begin{aligned} X &= 00,101011 \\ Y &= \underline{00,110100} \\ X+Y &= 01,011111 \end{aligned}$$

В ЭВМ в процессе работы оба знаковых разряда сравниваются. В случае появления признака переполнения машина останавливается.

Модифицированный дополнительный код также рассматривает два знаковых разряда, а во всем остальном ничем не отличается от обычного дополнительного кода, то есть:

1) для положительного числа

$$X = X_1 X_2 X_3 \dots X_n; \quad X_{доп}^M = 00, X_1 X_2 X_3 \dots X_n;$$

2) для отрицательного числа

$$X = X_1 X_2 X_3 \dots X_n; \quad X_{доп}^M = 11, \bar{X}_1 \bar{X}_2 \bar{X}_3 \dots \bar{X}_n + 0,000 \dots 1;$$

Пример. Даны два числа: $X=101001$ и $Y=-11010$. Сложить их в дополнительном и модифицированном дополнительном кодах.

1) Переведем X и Y в дополнительный и модифицированный дополнительный код:

Обычная запись	Обратный код	Дополнительный код
$X=+0101001$	$X_{обр}=0,0101001$	$X_{доп}=0,0101001$
$Y=-0011010$	$Y_{обр}=1,1100101$	$Y_{доп}=1,1100110$
Обычная запись	Мод. обратный код	Мод. дополнительный код
$X=+101001$	$X_{обр}^M=00,101001$	$X_{доп}^M=00,101001$
$Y=-011010$	$Y_{обр}^M=11,100101$	$Y_{доп}^M=11,100110$

2) Выполним сложение:

$$\begin{array}{rcl}
 X_{\text{доп}} = & 0,0101001 & X_{\text{доп}}^{\text{м}} = 00,101001 \\
 Y_{\text{доп}} = & \underline{1,1100110} & Y_{\text{доп}}^{\text{м}} = \underline{11,100110} \\
 & \swarrow 1)0,0001111 & \swarrow 1) 00,001111 \\
 & & \text{отбрасывается} \\
 \text{отбрасывается} & (X+Y)_{\text{доп}} = 0,0001111 & (X+Y)_{\text{доп}}^{\text{м}} = 00,001111
 \end{array}$$

Переполнения нет (в знаковых разрядах “00”), поэтому результаты, полученные в обычном и модифицированном кодах совпадают ($X+Y=1111$).

Упражнения 2.

1) Записать число в прямом, обратном и дополнительном кодах:

а) 11010; б) -11101; в) -101001; г) -1001110.

2) Перевести X и Y в прямой, обратный и дополнительный коды. Сложить их в обратном и дополнительном кодах. Результат перевести в прямой код. Проверить полученный результат, пользуясь правилами двоичной арифметики.

а) $X = -11010$; б) $X = -11101$; в) $X = 1110100$;
 $Y = 1001111$; $Y = -100110$; $Y = -101101$;

3)

г) $X = -10110$; д) $X = 1111011$; е) $X = -11011$;
 $Y = -111011$; $Y = -1001010$; $Y = -10101$.

Сложить X и Y в модифицированном обратном и модифицированном дополнительном восьмиразрядных кодах. В случае появления признака переполнения увеличить число разрядов в кодах и повторить суммирование. Результат перевести в прямой код и проверить, пользуясь правилами двоичной арифметики.

а) $X = 10110$; б) $X = 11110$; в) $X = -11010$;
 $Y = 110101$; $Y = -111001$; $Y = -100111$;

г) $X = -11001$; д) $X = -10101$; е) $X = -1101$;
 $Y = -100011$; $Y = 111010$; $Y = -111011$.

3. Формы представления чисел в ЭВМ.

При проектировании ЭВМ, создании инструментального и прикладного программного обеспечения разработчикам приходится решать вопрос о представлении в ЭВМ числовых данных. Для решения большинства прикладных задач обычно достаточно использовать целые и вещественные числа. Запись целочисленных данных в запоминающем устройстве ЭВМ не представляет затруднений: число переводится в двоичную систему и записывается в прямом коде. Диапазон

представляемых чисел в этом случае ограничивается количеством выделенных для записи разрядов. Для вещественных данных обычно используются две формы записи: число с фиксированной точкой (ЧФТ) и число с плавающей точкой (ЧПТ).

3.1 Числа с фиксированной точкой.

Форма записи числа с фиксированной точкой использовалась в основном на ранних этапах развития вычислительной техники. Запись числа с фиксированной точкой обычно имеет знаковый и цифровой разряды. Фиксированная точка означает, что на этапе конструирования ЭВМ было определено, сколько и какие разряды машинного слова отведены под изображение целой и дробной частей числа. Запятая в разрядной сетке может быть зафиксирована, в принципе, после любого разряда.

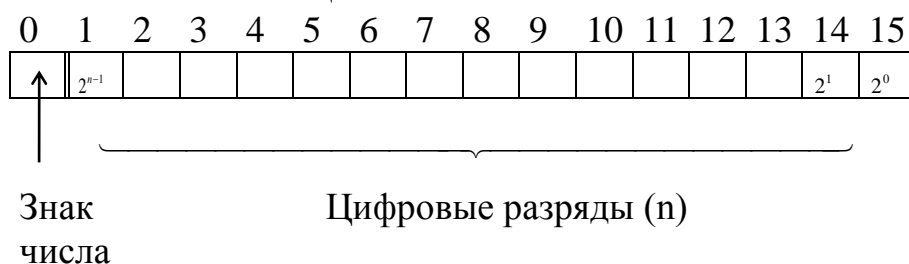
Пример.

Ячейка с целой и дробной частью.



Как частный случай числа с фиксированной точкой, может быть рассмотрена запись целого числа (в этом случае все разряды, кроме знакового, используются для записи целой части).

Ячейка с записью целого числа.



К достоинствам использования чисел с фиксированной точкой относятся простота выполнения арифметических операций и высокая точность изображения чисел. К недостаткам – небольшой диапазон представления чисел.

3.2 Числа с плавающей точкой.

Для представления чисел с плавающей точкой (ЧПТ) используется полулогарифмическая форма записи числа:

$$N = \pm m q^{\pm p}$$

где q – основание системы счисления, $\pm p$ – порядок числа, m – мантисса числа N .

Положение точки определяется значением порядка p . С изменением порядка точка перемещается (плавает) влево или вправо.

Пример.

$$125_{10} = 12.5 \cdot 10^1 = 1.25 \cdot 10^2 = 0.125 \cdot 10^3 = 0.0125 \cdot 10^4 = \dots$$

Для установления однозначности при записи чисел принята **нормализованная форма записи числа**. Мантисса нормализованного числа может изменяться в диапазоне: $1/q \leq |m| < 1$. Таким образом, в нормализованных числах цифра после точки должна быть значащей.

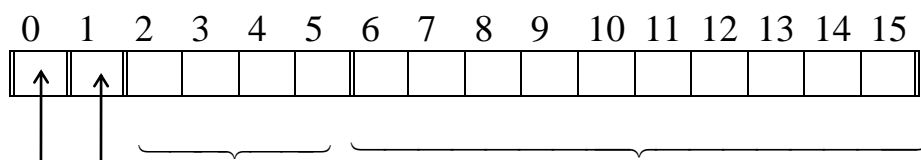
Пример.

$$\underbrace{0.0832 \cdot 10^3}_{\text{ненормализованное число}} = \underbrace{0.832 \cdot 10^2}_{\text{нормализованное число}}$$

ненормализованное число нормализованное число

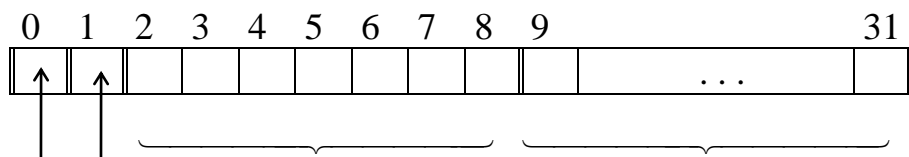
Для представления чисел в машинном слове выделяют группы разрядов для изображения мантиссы, порядка, знака числа и знака порядка:

а) представление чисел в формате полуслова



Знак m Знак p Порядок p (4 разряда) Мантисса m (10 разрядов)

б) представление чисел в формате слова



Знак m Знак p Порядок p (7 разрядов) Мантисса m (23 разряда)

Наиболее типично представление ЧПТ в формате слова (32 разряда).

Пример.

а) Число $A = 4_{10} = 100_2 = 0.1 \cdot 10^{11}$ записывается в ячейку следующим образом:



Знак m Знак p Порядок p (7 разрядов) Мантисса m (23 разряда)

б) Число $A = -3.5_{10} = -11.1_2 = -0.111 \cdot 10^{10}$



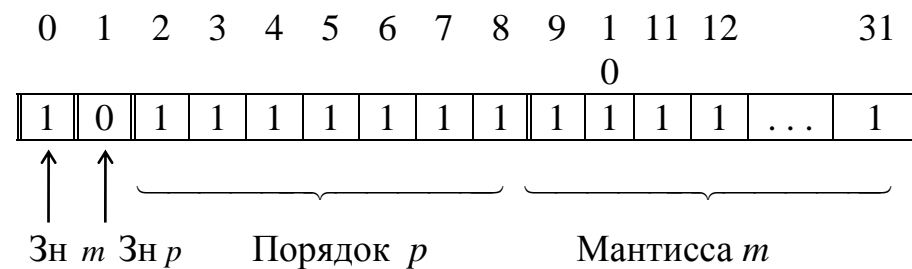
Максимальным числом представимым в формате слова будет

$$A = (\underbrace{0.111...1}_{23} \cdot 10^{1111111})_2 \cong (1 \cdot 2^{127})_{10}$$



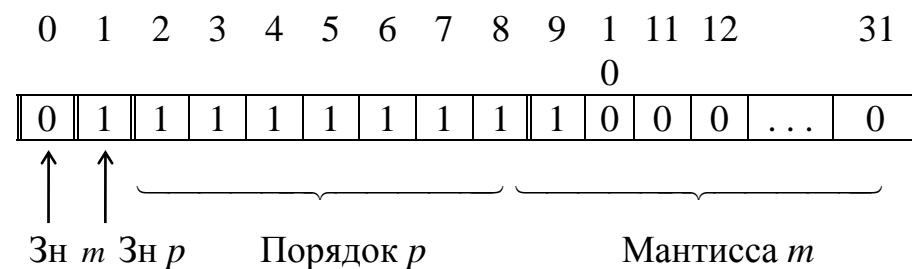
Минимальным числом из возможно представимых в формате слова будет

$$A = (-\underbrace{0.111...1}_{23} \cdot 10^{1111111})_2 \cong (-1 \cdot 2^{127})_{10}$$



Минимальным по модулю, отличным от нуля и нормализованным будет

$$A = (0.1 \cdot 10^{-1111111})_2 = (\frac{1}{2} \cdot 2^{-127})_{10} = (2^{-128})_{10}$$



Таким образом, числа с плавающей точкой позволяют увеличить диапазон обрабатываемых чисел, но при этом точность изображения чисел определяется только разрядами мантиссы и уменьшается по сравнению с числами с фиксированной точкой. При записи числа в формате слова диапазон представимых чисел будет от $-1 \cdot 2^{127}$ до $1 \cdot 2^{127}$ ($2^{127} \approx 10^{38}$), а точность определяться мантиссой, состоящей из 23 разрядов. Точность может быть повышена путем увеличения количества разрядов мантиссы. Это реализуется путем представления чисел с так называемой двойной точностью (используется формат двойного слова):



Литература.

1. Пономарев В.С., Красников В.В. Методические указания по курсу "Организация и функционирование ЭВМ и систем". Ч. 1. Арифметические основы ЭВМ. ДГТУ, 1996.
2. Каган Б.М. Электронные вычислительные машины и системы. М.: Энергоатомиздат, 1991.
3. Савельев А.Я. Прикладная теория цифровых автоматов. М.: Высшая школа, 1983.
4. Лю Ю-Чжен, Гибсон Г. Микропроцессоры семейства 8086/8088. М.: Радио и связь