

Кодирование числовой
информации.

Системы счисления.

Представление чисел в
компьютере.

Системы счисления, применяемые для представления числовых данных в ЭВМ

Под системой счисления понимается способ представления любого числа посредством некоторого алфавита символов, называемых цифрами.

Как известно, системы счисления (СС) бывают позиционные и непозиционные.

В позиционной системе счисления в зависимости от положения (разряда) в котором находится число оно имеет разное значение. Например: 123 ("1" - сотни, "2" - десятки, "3" - единицы)

В непозиционных системах счисления число не меняет своего значения в зависимости от позиции. Например: XXV, XVI, VII (V везде значит – 5)

? Числовые данные обрабатываются в компьютере в двоичной системе счисления. Числа хранятся в оперативной памяти в виде последовательностей нулей и единиц, т.е. в двоичном коде.

Системы счисления, применяемые для представления числовых данных в ЭВМ

В позиционной системе счисления числа записываются в виде последовательности цифр:

$$A = a_{m-1} a_{m-2} \dots a_1 a_0, a_{-1} a_{-2} a_{-3} \dots a_{-n}. \quad (1)$$

Записанную выше последовательность цифр (1), соответствующую числу A , можно представить в виде полинома (2) от основания q :

$$A = a_{m-1} q^{m-1} + a_{m-2} q^{m-2} + \dots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + a_{-2} q^{-2} + \dots + a_{-n} q^{-n} \quad (2)$$

Основание системы счисления определяет ее название, например, $q = 10$ – десятичная система счисления, а $q = 2$ – двоичная.

В ЭВМ применяют позиционные системы счисления с недесятичным
4
основанием: двоичную, восьмеричную, шестнадцатеричную.

Системы счисления, применяемые для представления числовых данных в ЭВМ

Принятые обозначения:

двоичная СС - $(A)_2$,
десятичная СС - $(A)_{10}$,
восьмеричная СС - $(A)_8$,
шестнадцатеричная СС - $(A)_{16}$.

Позиционные системы счисления

Основание системы равно количеству цифр (знаков) в ее алфавите

Система счисления	Основание	Алфавит цифр
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двоичная	2	0, 1
Восьмеричная	8	0, 1, 2, 3, 4, 5, 6, 7
Шестнадцатеричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Позиция цифры в числе называется **разрядом**

Соответствие систем счисления

Десятичная	0	1	2	3	4	5	6	7
Двоичная	0	1	10	11	100	101	110	111
Восьмеричная	0	1	2	3	4	5	6	7
Шестнадцатеричная	0	1	2	3	4	5	6	7

Десятичная	8	9	10	11	12	13	14	15	16
Двоичная	1000	1001	1010	1011	1100	1101	1110	1111	10000
Восьмеричная	10	11	12	13	14	15	16	17	20
Шестнадцатеричная	8	9	A	B	C	D	E	F	10

Системы счисления, применяемые для представления числовых данных в ЭВМ

В двоичной системе счисления любое число в соответствии с (1) и (2) может быть представлено последовательностью двоичных цифр (3) или суммой степеней числа 2, взятых с указанными в ней коэффициентами (4).

$$X = a_{m-1} a_{m-2} \dots a_1 a_0, a_{-1} a_{-2} a_{-3} \dots, \quad (3)$$

где $a_i = \{0,1\}$;

$$X = a_{m-1} * 2^{m-1} + \dots + a_1 * 2^1 + a_0 * 2^0 + a_{-1} * 2^{-1} + a_{-2} * 2^{-2} + \dots + a_{-n} * 2^{-n} \quad (4)$$

Например, двоичное число 1010,001 будет представлено следующим образом:

$$(1110,001)_2 = 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 + 0 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3}$$

8

Системы счисления, применяемые для представления числовых данных в ЭВМ

В восьмеричной системе счисления используется восемь цифр: 0,1,2,3,4,5,6,7. Любое число в восьмеричной системе может быть представлено последовательностью цифр или суммой степеней числа 8.

$$(A)_8 = (157,34)_8 = 1 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 + 3 \cdot 8^{-1} + 4 \cdot 8^{-2}$$

В шестнадцатеричной системе счисления для изображения чисел употребляются 16 цифр от 0 до 15. При этом, чтобы одну цифру не изображать двумя знаками, введены обозначения для цифр, больших девяти, латинскими буквами: десять – А, одиннадцать – В, двенадцать – С, тринадцать – D, четырнадцать – E, пятнадцать – F.

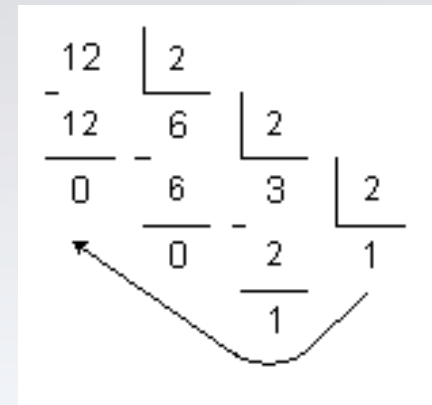
9

$$(A)_{16} = (2AF)_{16} = 2 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0$$

Перевод целых чисел

Для перевода целого числа A , представленного в системе счисления с основанием q , в систему счисления с основанием S необходимо данное число и получаемые частные последовательно делить на новое основание системы счисления до тех пор, пока последнее частное не будет меньше S . Число A в системе счисления с основанием S представится в виде упорядоченной последовательности остатков деления, причем старшую цифру дает последнее частное.

$$(12)_{10} = (1100)_2$$



Перевод дробных чисел

Перевод дробных чисел заключается в последовательном умножении дроби на основание новой системы счисления, причем перемножению подвергаются только дробные части результата. Дробь в новой системе счисления представляется в виде упорядоченной последовательности целых частей произведений, где старший разряд является первой цифрой произведения.

$$(0,325)_{10} = (0,0101)_2$$

	0,325
	<u> </u>
	2

	0,650
	<u> </u>
	2

	1,300
	<u> </u>
	2

	0,600
	<u> </u>
	2

↓	1,200

ПЕРЕВОД $(A)_8 \longrightarrow (A)_2$

Для перевода восьмеричного числа в двоичное достаточно каждую цифру числа заменить трехразрядным двоичным числом.

При этом отбрасывают нули, стоящие слева от старшей значащей цифры и справа от младшей значащей цифры двоичного кода.

$$\underbrace{1}_{001} \quad \underbrace{7}_{111} \quad \underbrace{5}_{101}, \quad \underbrace{6}_{110}; (175,6)_8 = (1111101,11)_2$$

$$1 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 + 6 \cdot 8^{-1} = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

$$(175,6)_8 = (125,75)_{10}, \quad (1111101,11)_2 = (125,75)_{10}$$

ПЕРЕВОД $(A)_{16} \rightarrow (A)_2$

Для перевода шестнадцатеричного числа в двоичное достаточно заменить каждую цифру числа четырехразрядным двоичным кодом.

<u>2</u>	<u>C</u>	<u>F</u>	<u>5</u>
0010	1100	1111	0101

$$(2CF,5)_{16} = (1011001111,0101)_2$$

ПЕРЕВОД $(A)_2 \longrightarrow (A)_8$ и $(A)_2 \longrightarrow (A)_{16}$

- перевод двоичного числа 110101,01 в восьмеричное:

$$\underbrace{110}_6 \quad \underbrace{101}_5 \quad , \quad \underbrace{010}_2 = (65,2)_8$$

- перевод двоичного числа 111000110,101 в шестнадцатеричное

$$\underbrace{0001}_1 \quad \underbrace{1100}_C \quad \underbrace{0110}_6 \quad , \quad \underbrace{1010}_A = (1C6,A)_{16}$$

Представление двоичных чисел в форме с фиксированной запятой

При представлении чисел с фиксированной запятой положение запятой (точки) фиксировано относительно разрядов числа и сохраняется неизменным для всех чисел.

Запятая отделяет целую часть числа от дробной.

Если дробная часть отсутствует, то число – целое.

Для кодирования знака используется знаковый разряд («0» для положительных чисел и «1» – для отрицательных).

Представление чисел в формате с фиксированной запятой

Целые числа в компьютере хранятся в памяти в формате с фиксированной запятой. В этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа, а запятая находится справа после младшего разряда, т.е. вне разрядной сетки.

1	0	1	0	1	0	1	0
---	---	---	---	---	---	---	---

Для хранения *целых неотрицательных чисел* отводится одна ячейка памяти (8 бит). Например, число $A_2 = 10101010_2$ будет храниться в ячейке памяти следующим образом:

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех ячейках хранятся единицы. Для n -разрядного представления оно будет равно:

$$2^n - 1$$

Для хранения *целых чисел со знаком* отводится две ячейки памяти (16 бит), причем старший (левый) разряд отводится под знак числа (если число положительное, то в знаковый разряд записывается 0, если число отрицательное записывается 1).

Представление в компьютере положительных чисел с использованием формата «знак-величина» называется *прямым кодом* числа. Например, число $2002_{10} = 11111010010_2$ будет представлено в 16-ти разрядном представлении следующим образом:

0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Для получения дополнительного кода отрицательного числа можно использовать довольно простой алгоритм:

1. Модуль числа записать *прямым кодом* в n двоичных разрядах;
2. Получить *обратный код* числа, для этого значения всех бит инвертировать (все единицы заменить на нули и все нули заменить на единицы);
3. К полученному *обратному коду* прибавить единицу.

Пример. Записать дополнительный код отрицательного числа -2002 для 16-ти разрядного компьютерного представления с использованием алгоритма.

Прямой код	$ -2002_{10} $	0000011111010010 ₂
Обратный код	инвертирование	1111100000101101 ₂
	прибавление единицы	1111100000101101 ₂ +0000000000000001 ₂
Дополнительный код		1111100000101110 ₂

Представление двоичных чисел в форме с фиксированной запятой

Если для представления числа со знаком выделено n разрядов, то диапазон представления целых двоичных чисел в этом случае определяется выражением

$$1 \leq |X_{\text{ф.з.}}^{\text{ц}}| \leq 2^{n-1} - 1.$$

Диапазон представления в ЭВМ дробных двоичных чисел будет определяться неравенством

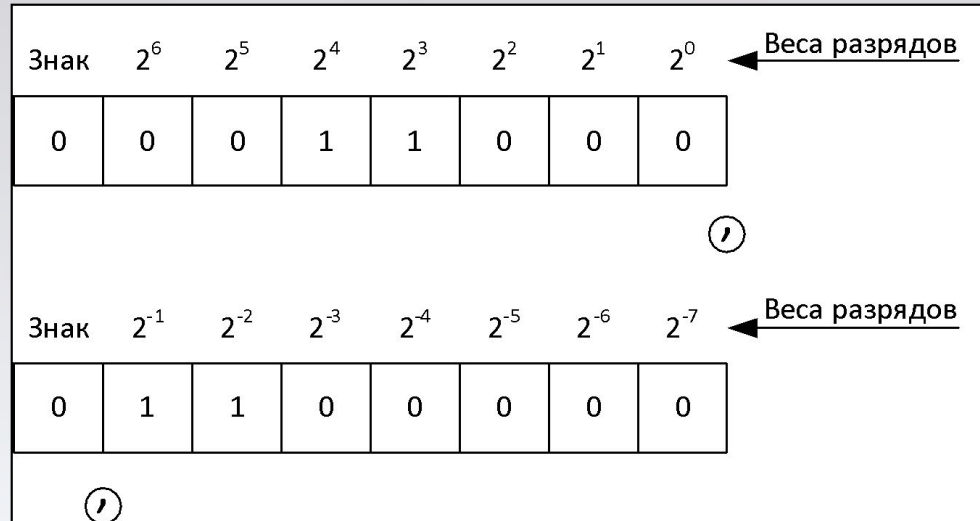
$$2^{-(n-1)} \leq |X_{\text{ф.з.}}^{\text{др.}}| \leq 1 - 2^{-(n-1)}$$

или приближенно

$$0 \leq |X_{\text{ф.з.}}| < 1.$$

Представление двоичных чисел в форме с фиксированной запятой

Разрядная сетка ЭВМ в формате 8-разрядного машинного слова для представления соответственно целого двоичного числа ($= +11000$) и дробного числа ($= +0,11$) в форме с фиксированной запятой:



Представление двоичных чисел в форме с фиксированной запятой

Пусть задано число $(X)_2 = -100010$.

- Целое число $(X)_2$ в формате (n=7 со знаком):

Знак	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	0	1	0

- Целое число $(X)_2$ в формате (n=8 со знаком):

Знак	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	0	0	1	0

- Дробное число $(X)_2$ в формате (n=8 со знаком):

Знак	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
1	1	0	0	0	1	0	0

Представление двоичных чисел в форме с плавающей запятой

Вещественные числа (конечные и бесконечные десятичные дроби) хранятся и обрабатываются в компьютере в формате *с плавающей запятой*. В этом случае положение запятой в записи числа может изменяться.

Формат чисел *с плавающей запятой* базируется на экспоненциальной форме записи, в которой может быть представлено любое число. Так число A может быть представлено в виде:

$$A = m \times q^n$$

где m – мантисса числа

q – основание системы счисления,

n – порядок числа.

Для однозначности представления чисел *с плавающей запятой* используется нормализованная форма, при которой мантисса отвечает условию:

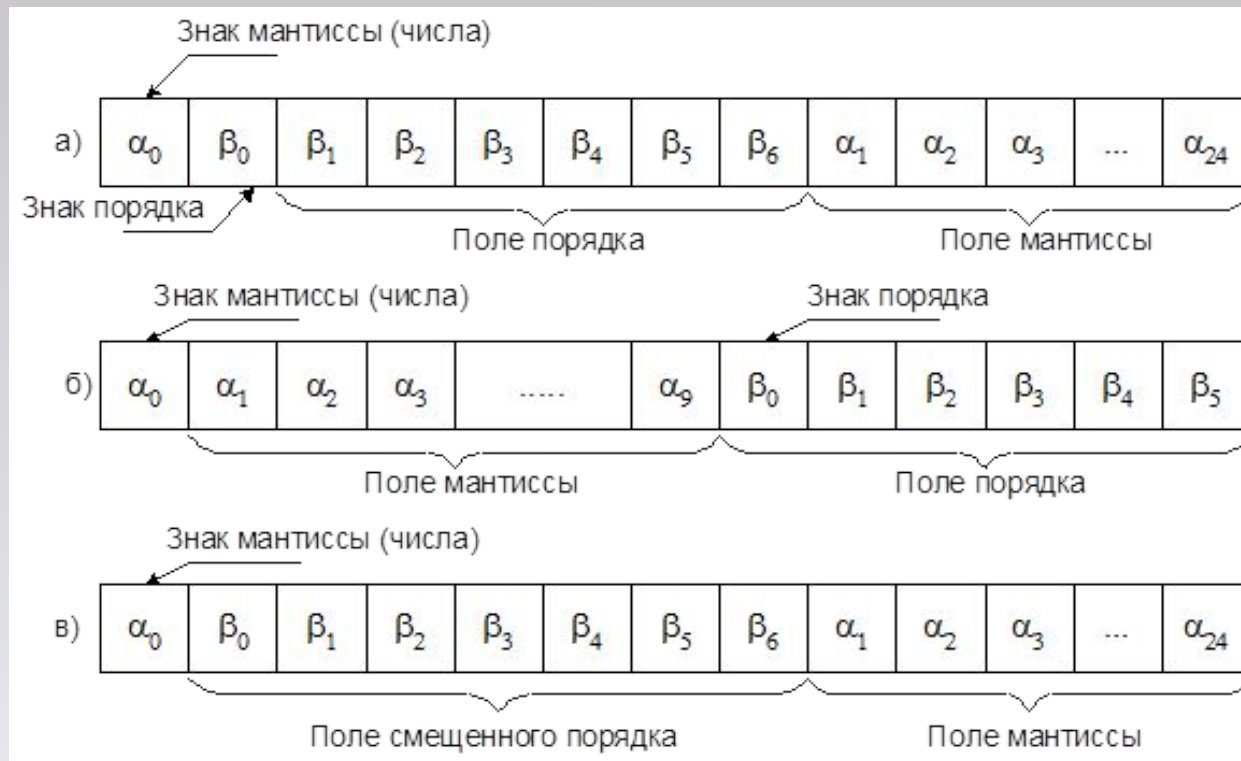
$$1/n \leq |m| < 1.$$

Это означает, что мантисса должна быть правильной дробью и иметь после запятой цифру, отличную от нуля.

Представление двоичных чисел в форме с плавающей запятой

Запятая при представлении мантиссы фиксируется перед старшим значащим разрядом. Порядок P указывает положение запятой в числе, может быть положительным или отрицательным целым числом или целым числом без знака (запятая при представлении порядка фиксируется после младшего разряда). Порядок P и мантисса m_n представляются в системе счисления с основанием q .

Форматы представления в ЭВМ чисел с плавающей запятой



Кодирование отрицательных чисел в ЭВМ

Прямой код чисел соответствует обычной записи чисел со своим знаком:

$$A_1 = +0,0101, \quad [A_1]_{\text{пр}} = 00101;$$

$$A_2 = -0,0101, \quad [A_2]_{\text{пр}} = 10101.$$

Для целых чисел в двоичной системе счисления:

$$A_1 = +1100, \quad [A_1]_{\text{пр}} = 01100;$$

$$A_2 = -1100, \quad [A_2]_{\text{пр}} = 11100.$$

Нуль в прямом коде имеет два изображения:

$$+0 = 000\dots00 = [0]_{\text{пр}'}$$

$$-0 = 100\dots00 = [0]_{\text{пр}}$$

Кодирование отрицательных чисел в ЭВМ

Обратный код. Чтобы представить двоичное отрицательное число в обратном коде, нужно поставить в знаковый разряд единицу, а все остальные разряды инвертировать:

$$A = -0,1010. [A]_{\text{обр}} = 10101.$$

Примеры обратного кода отрицательных дробного и целого чисел:

$$A^{\text{др}} = -0,11001;$$

$$[A^{\text{др}}]_{\text{пр}} = 111001;$$

$$[A^{\text{др}}]_{\text{обр}} = 100110;$$

$$A^{\text{ц}} = -10101;$$

$$[A^{\text{ц}}]_{\text{пр}} = 110101;$$

$$[A^{\text{ц}}]_{\text{обр}} = 101010;$$

Кодирование отрицательных чисел в ЭВМ

Дополнительный код. Для представления отрицательного числа в дополнительном коде нужно поставить единицу в знаковом разряде, затем найти крайнюю правую единицу и заменить на противоположные разряды слева (до знака). Остальное не менять.

Примеры:

$$[A]_{\text{пр}}^{\text{дп}} = 1110010; [A]_{\text{доп}}^{\text{дп}} = 1001110 ,$$

$$[A]_{\text{пр}}^{\text{ц}} = 1001110; [A]_{\text{доп}}^{\text{ц}} = 1110010 ,$$

$$[A]_{\text{пр}}^{\text{ц}} = 1001001; [A]_{\text{доп}}^{\text{ц}} = 1110111 .$$

Кодирование отрицательных чисел в ЭВМ

Правило перевода отрицательных чисел из обратного кода в дополнительный:

дополнительный код отрицательного числа может быть получен из обратного путем прибавления к нему единицы младшего разряда.

Примеры:

$$[A]_{\text{пр}} = 1.01010;$$

$$[A]_{\text{пр}} = 1.11101;$$

$$[A]_{\text{обр}} = 110101;$$

$$[A]_{\text{обр}} = 100010;$$

$$[A]_{\text{доп}} = 110110,$$

$$[A]_{\text{доп}} = 100011.$$

Отрицательный нуль изображается:

- в обратном коде $[-0]_{\text{обр}} = 1.11111\dots 11;$

- в дополнительном коде отрицательный нуль отсутствует, т.е. код нуля в ²⁸дополнительном коде соответствует коду нуля положительного числа.

Кодирование отрицательных чисел в ЭВМ

**Положительные числа в прямом,
обратном и дополнительных
кодах имеют
одинаковую форму записи!!!**

Кодирование отрицательных чисел в ЭВМ

Модифицированный код. При выполнении арифметических операций в ЭВМ иногда возникает необходимость для представления знака числа использовать не один, а два знаковых разряда.

Модифицированный код отличается от обычного двумя разрядами для знака. Примеры:

$$\begin{aligned} [A_1]_{\text{пр.}} &= 1\ 01001, & [A_1]_{\text{пр. мод.}} &= 11\ 01001, \\ [A_1]_{\text{обр.}} &= 1\ 10110, & [A_1]_{\text{обр. мод.}} &= 11\ 10110, \\ [A_3]_{\text{доп.}} &= 1\ 10111, & [A_3]_{\text{доп. мод.}} &= 11\ 10111. \end{aligned}$$