

# СИСТЕМЫ СЧИСЛЕНИЯ. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В КОМПЬЮТЕРЕ

Зав. кафедрой, к.ф.-м.наук

Тишков Артем Валерьевич

доцент, к.п.н.

Никонорова Маргарита Леонидовна

2019



*«Письменность, это величайшее изобретение руки и ума человека, постепенно возникла из счета»*

Дж. Бернал, английский ученый и философ

## **Система счисления**

– это способ отображения чисел с помощью символов или совокупность правил именования и изображения чисел с помощью набора символов, называемых цифрами.

Используются три типа систем счисления:

- **позиционная** – представление числа зависит от порядка записи цифр.
- **непозиционная** – представление числа не зависит от порядка записи цифр
- **смешанная** – нет понятия «основание»: либо оснований несколько, либо оно вычисляемое



# НЕПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

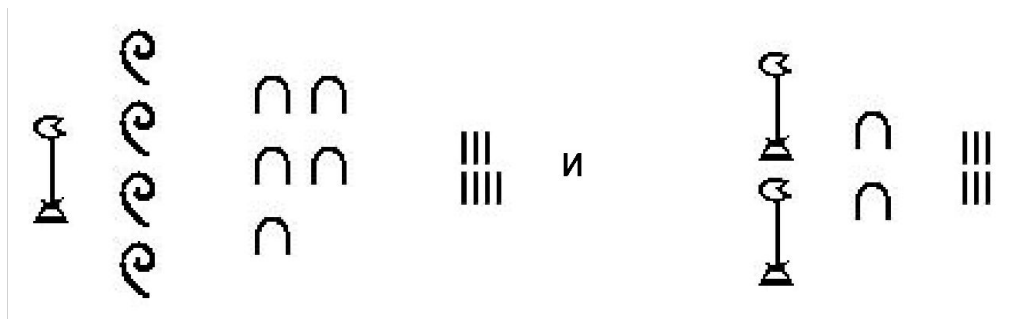
## Единичная (палочная)

В этой системе счисления для записи чисел используется только одна цифра. Ее можно изобразить в виде палочки |, кружочка ⊗, или любой другой фигуры. Числа записываются примерно так:

1 2 3  
| || |||

## Древнеегипетская десятичная

I	II	III	IIII		IIII IIII	∩	∩
1	2	3	4	...	9	10	11



$$1457 \times 2026$$

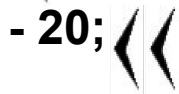


## Вавилонская нумерация

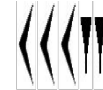
В вавилонской нумерации основную роль играет число 60, и потому эту нумерацию называют шестидесятиричной.

Числа менее 60 обозначались с помощью двух знаков:

▼ для единицы, и ◁ для десятка.



- 32



Вавилонский способ обозначения числа 60 снова обозначался ▼ знаком тем же, что и единица.

Однако отсутствие низшего разряда не обозначалось, поэтому число  $180 = 3 \times 60$  записывалось так ▼▼▼, а обозначать эта запись могла и 3, и 180, и 10800 ( $3 \times 60 \times 60$ ) и т.д. Различать эти числа можно было только по смыслу текста.

## Римская нумерация

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Записывались цифры числа начиная с больших значений и заканчивая меньшими, слева направо. При записи цифры с меньшим значением перед цифрой с большим значением происходит ее вычитание.

ССХХХVII =  $100+100+10+10+10+5+1+1 = 237$ , но ХХХIX =  $10+10+10-1+10 = 39$

*Правило:* нельзя записывать подряд 4 одинаковых цифры, такая комбинация заменяется комбинацией с правилом вычитания, например: ХХХХ = ХС ( $50-10$ )

III = IV ( $5-1$ ),      CCCC = CD ( $500-100$ )



## Славянская кириллическая нумерация

Эта нумерация была создана для переписки священных книг для славян греческими монахами братьями **Кириллом (Константином) и Мефодием в IX веке**. До XVII века эта форма записи чисел была официальной на территории современной **России, Белоруссии, Болгарии, Венгрии, Сербии и Хорватии**. До сих пор православные церковные книги используют эту нумерацию.

а - 1	І - 10	ρ - 100
в - 2	к - 20	с - 200
г - 3	л - 30	т - 300
д - 4	м - 40	ϥ - 400
е - 5	н - 50	φ - 500
ѕ - 6	ѣ - 60	χ - 600
з - 7	о - 70	ψ - 700
и - 8	п - 80	ω - 800
ѹ - 9	ч - 90	ц - 900

Интереснее всего записывались числа второго десятка: **ДІ - 14**

Читаем дословно «четыре на десять» – «четыре на десять» 4+10, и так для всех чисел от 11 до 19.

$\overline{\omega\zeta\Gamma}$  - 863 800+60+3.

Для того, чтобы не перепутать буквы и цифры, использовалась титла - горизонтальная черточка над числами.

Для обозначения больших, чем 900 чисел использовались специальные значки, добавляемые к букве, буква бралась в кружочек.

# Китайская нумерация



Возникла эта нумерация около 4 000 тысяч лет тому назад в Китае.

一	二	三	四	五	六	七	八	九	〇
1	2	3	4	5	6	7	8	9	0

Использовали несколько служебных иероглифов, чтобы не перепутать разряды.

十	百	千
10	100	1000

一	千	五	百	四	十	八
	- 1 000;					- 548

Такая запись числа мультипликативна:

$$1 \times 1\,000 \text{ и } 5 \times 100 + 4 \times 10 + 8$$



## Греческая нумерация

В древнейшее время в Греции была распространена так называемая Аттическая нумерация. В этой нумерации числа 1, 2, 3, 4 изображались соответствующим количеством вертикальных полосок: |, ||, |||, ||||. Число 5 записывалось знаком  $\Gamma$  (древнее начертание буквы «Пи», с которой начиналось слово «пять» - «пенте»). Числа 6, 7, 8, 9 обозначались сочетаниями этих знаков:

$\Gamma$ ,  $\Gamma\Gamma$ ,  $\Gamma\Gamma\Gamma$ ,  $\Gamma\Gamma\Gamma\Gamma$

Число 10 обозначалось  $\Delta$  – заглавной «Дельта» от слова «дека» – «десять». Числа 100, 1 000 и 10 000 обозначались H, X, M. Числа 50, 500, 5 000 обозначались комбинациями чисел 5 и 10, 5 и 100, 5 и 1 000.

Примерно в третьем веке до нашей эры аттическая нумерация в Греции была вытеснена другой, так называемой «Ионийской» системой.

В ней числа 1 – 9 обозначаются первыми буквами греческого алфавита:

$\alpha = 1, \beta = 2, \gamma = 3, \delta = 4, \varepsilon = 5, \varsigma = 6, \zeta = 7, \eta = 8, \vartheta = 9$

числа 10, 20, ... 90 изображались следующими девятью буквами;

числа 100, 200, ... 900 последними девятью буквами.

Примерно по такому же принципу организованную систему счисления имели в древности евреи, арабы и многие другие народы Ближнего Востока.



# Непозиционные системы счисления

Представление через

биномиальные коэффициенты

$$x = \sum_{k=1}^n \binom{c_k}{k} \quad 0 \leq c_1 < c_2 < \dots < c_n$$

## Система остаточных классов

Определяется набором взаимно простых модулей  $(m_1, m_2, \dots, m_n)$  с произведением  $M = m_1 * m_2 * \dots * m_n$  так, что каждому целому числу  $x$  из отрезка  $[0, M - 1]$  ставится в соответствие набор  $(x_1, x_2, \dots, x_n)$  вычетов

$$x \equiv x_1 \pmod{m_1};$$

$$x \equiv x_2 \pmod{m_2};$$

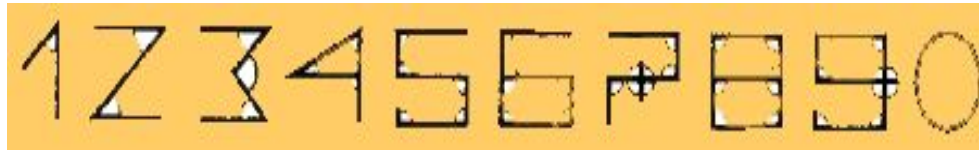
$$x \equiv x_n \pmod{m_n};$$





# ПОЗИЦИОННЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Системы счисления, основанные на **позиционном принципе**, возникли независимо друг от друга в трех местах: в **Древнем Вавилоне**, у племени **Майя** и в **Индии**.



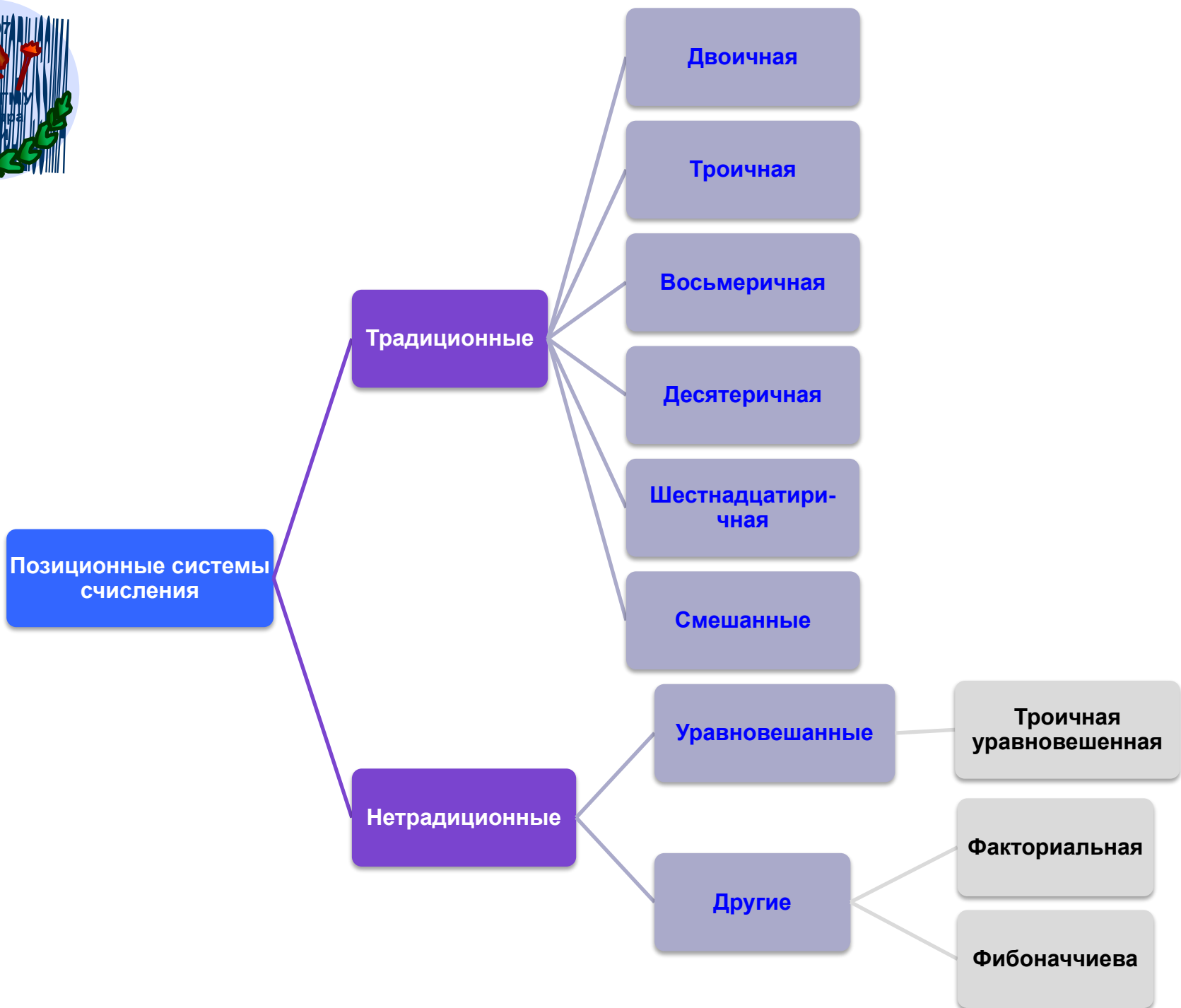
Каждая цифра обозначает число по количеству углов в ней. Например, 0 – углов нет, 1 – один угол, 2 – два угла и т.д.

## Появление нуля

Цифрой 0 обозначается отсутствующая величина.

Древние греческие астрономы (II век до н.э.) для обозначения нулевого значения разряда использовали символ 0 (по первой букве греческого слово **ничто**). Этот знак и был прообразом современного нуля. Индийцы переняли общетеоретические положения этой науки и многие греческие термины.

Соединив свою десятичную мультипликативную систему с принципами нумерации чисел греческих астрономов, индийские ученые сделали завершающий шаг в создании всем известной десятичной системы счисления.





Известный математик и физик XVIII–XIX века П. Лаплас говорил:

*«Мысль выразить все числа десятью знаками, придавая им, кроме значения по форме, еще значение по месту, настолько проста, что именно из-за этой простоты трудно понять, насколько она удивительна. Как нелегко было прийти к этому методу, мы видим на примере величайших гениев греческой учености Архимеда и Апполония, от которых эта мысль осталась скрытой.»*

## Позиционная, двоичная

- Логика: истина / ложь
- В повседневной жизни: Да / Нет
- В повседневной жизни: Есть / Нет
- В технике: электрический сигнал есть / нет
- 0 / 1, бит



## Позиционная, двоичная

В двоичной системе счисления используются всего 2 состояния или 2 цифры. Такое представление информации принято считать двоичным кодированием.

Великий немецкий математик Г.Ф. Лейбниц видел в ней «...прообраз творения». Он считал, что «единица представляет божественное начало, а ноль – небытие. Высшее существо создает все сущее из небытия точно таким же образом, как единица с помощью нуля выражает все числа».

Вся информация, обрабатываемая компьютерами хранится в двоичном виде в ячейках памяти, а обрабатывается – в регистрах арифметико–логического устройства (АЛУ). И регистр и ячейка памяти представляет собой физическую систему, состоящую из некоторого числа однородных элементов, которые называются **разрядом**.

Нумерация справа → налево:

( $n-1$ )-й разряд

0-й разряд



ячейка из  $n$ -разрядов



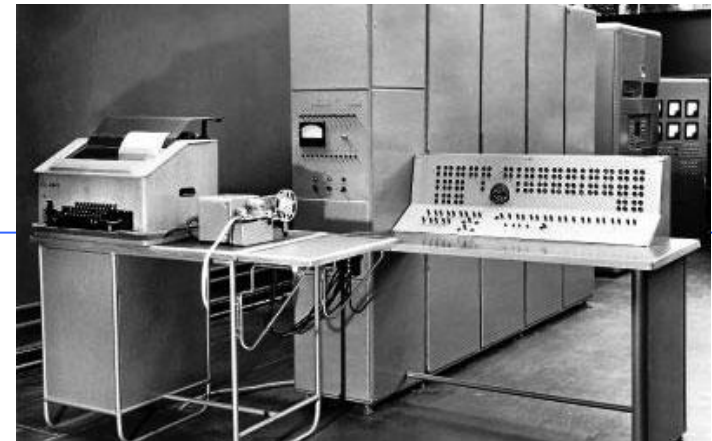
# Позиционная, троичная. Симметричная и несимметричная

Десятичная система	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
Троичная несимметричная	-100	-22	-21	-20	-12	-11	-10	-2	-1	0	1	2	10	11	12	20	21	22	100
Троичная симметричная	$\bar{1}00$	$\bar{1}01$	$\bar{1}1\bar{1}$	$\bar{1}10$	$\bar{1}11$	$\bar{1}\bar{1}$	$\bar{1}0$	$\bar{1}1$	$\bar{1}$	0	1	$1\bar{1}$	10	11	$1\bar{1}\bar{1}$	$1\bar{1}0$	$1\bar{1}1$	$10\bar{1}$	100

- **Трит** – троичный триггер
- 1 **Трайт** = 6 тритов, 729 значений (байт – 256)
- Советская машина Сетунь – первая и единственная серийная троичная машина. 1962-1965 годы

$$10\bar{1} = 9 - 1 = 8$$

$$\bar{1}01 = -9 + 1 = -8$$





# Преимущества троичной симметричной системы $(-1, 0, 1)$

- естественное представление чисел со знаком (не нужен прямой, обратный или дополнительный код!)
- знак числа - это знак старшей ненулевой цифры и не нужен знаковый бит
- Простое сравнение чисел по величине, при этом не нужно смотреть на знак
  - поэтому команда ветвления по знаку в троичной машине работает вдвое быстрее, чем в двоичной
- усечение длины числа равносильно правильному округлению (округление в двоичных машинах не обеспечивают этого)
- троичный сумматор осуществляет вычитание при инвертировании одного из слагаемых, откуда следует, что троичный счетчик автоматически является реверсивным (обеспечивает и сложение и вычитание)
- в трехвходовом троичном сумматоре перенос в следующий разряд возникает в 8 ситуациях из 27, а в двоичном сумматоре - в 4 из 8. В четырехвходовом сумматоре перенос также происходит только в соседний разряд.
- таблицы умножения и деления почти так же просты, как и в двоичной системе
- умножение на  $-1$  инвертирует множимое
- трехуровневый сигнал более устойчив к воздействию помех в линиях передачи. Это означает что специальные методы избыточного кодирования троичной информации проще, нежели двоичной



## Позиционные системы счисления

- Десятичная
- Двоичная
- Восьмеричная
- Шестнадцатеричная
- Троичная (электроника +,0,-)
- Двенадцатеричная (счет дюжинами)
- Шестидесятеричная (время, углы – широта долгота)

Количество цифр называют **основанием** позиционной системы счисления, а позиции цифр в числе – **разрядами**.

### Алфавит

- Двоичная система счисления – 0, 1
- Восьмеричная система счисления – 0, 1, 2, 3, 4, 5, 6, 7
- Шестнадцатеричная система счисления –  
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F  
10, 11, 12, 13, 14, 15



# Таблица шестнадцатеричных чисел

$X_{10}$	$X_{16}$	$X_2$	$X_{10}$	$X_{16}$	$X_2$
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111





# Смешанные системы счисления

**Дата** – год, месяц, день

**Время** – часы, минуты, секунды, миллисекунды

при этом величина  $h$  часов,  $m$  минут,  $s$  секунд соответствует

$$T = h * 60 * 60 + m * 60 + s$$

**Углы** – градусы, минуты, секунды

**Смешанной** называется система счисления, в которой числа, заданные в некоторой системе счисления с основанием  $P$  (например – время) изображаются с помощью цифр другой системы счисления с основанием  $Q$  (например – секунды), где  $Q < P$ .

В такой системе  $P$  называется **старшим основанием**,  $Q$  – **младшим основанием**, а сама система счисления называется  **$Q$ – $P$ –ичной**.



# Представление числа в позиционной системе счисления

$$a_{n-1}a_{n-2} \dots a_1a_0, a_{-1} \dots a_{-m}$$

Запись чисел в каждой из систем счисления с основанием  $q$  означает сокращенную запись выражения

$$a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_1q^1 + a_0q^0 + a_{-1}q^{-1} + \dots + a_{-m}q^{-m},$$

где  $a_i$  – цифры численной записи, соответствующие разрядам,  $i$  – индекс,  $n$  и  $m$  – количество разрядов числа целой и дробной части соответственно,  $q$  – основание системы счисления.

Например, развернутая форма числа 327,46

$$n = 3, m = 2, q = 10$$

$$X = \sum_{i=2}^{-2} a_i q^i, *10^2 + a_1 *10^1 + a_0 *10^0 + a_{-1} *10^{-1} + a_{-2} *10^{-2} =$$

$$3*10^2 + 2*10^1 + 7*10^0 + 4*10^{-1} + 6*10^{-2}$$



## Перевод целых чисел из десятичной системы счисления

Пример. Перевести число 75 из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную.

$\begin{array}{r} -75 \mid 2 \\ \hline 74 \\ \hline 1 \end{array}$	$\begin{array}{r} -37 \mid 2 \\ \hline 36 \\ \hline 1 \end{array}$	$\begin{array}{r} -18 \mid 2 \\ \hline 18 \\ \hline 0 \end{array}$	$\begin{array}{r} -9 \mid 2 \\ \hline 8 \\ \hline 1 \end{array}$	$\begin{array}{r} -4 \mid 2 \\ \hline 4 \\ \hline 0 \end{array}$	$\begin{array}{r} -2 \mid 2 \\ \hline 2 \\ \hline 0 \end{array}$	$\begin{array}{r} -2 \mid 2 \\ \hline 1 \\ \hline 1 \end{array}$
--	--	--	--	--	--	--

$\begin{array}{r} -75 \mid 8 \\ \hline 72 \\ \hline 3 \end{array}$	$\begin{array}{r} -9 \mid 8 \\ \hline 8 \\ \hline 1 \end{array}$	$\begin{array}{r} -8 \mid 8 \\ \hline 8 \\ \hline 1 \end{array}$
--	--	--

$75_{10} = 113_8$

$\begin{array}{r} -75 \mid 16 \\ \hline 64 \\ \hline 11 \end{array}$	$\begin{array}{r} -4 \mid 16 \\ \hline 4 \\ \hline 4 \end{array}$
--	---

$75_{10} = 4B_{16}$

$75_{10} = 1001011_2$



## Перевод правильной десятичной дроби из десятичной системы счисления

*Пример.* Перевести число 0,35 из десятичной системы в счисления в двоичную, восьмеричную и шестнадцатеричную.

$$\begin{array}{r} \times 0,35 \\ \hline 2 \\ \times 0,70 \\ \hline 2 \\ \times 1,40 \\ \hline 2 \\ \times 0,80 \\ \hline 2 \\ \times 1,60 \\ \hline 2 \\ \hline 1,20 \end{array}$$

$$0,35_{10} = 0,01011_2$$

$$\begin{array}{r} \times 0,35 \\ \hline 8 \\ \times 2,80 \\ \hline 8 \\ \times 6,40 \\ \hline 8 \\ \times 3,20 \\ \hline 8 \end{array}$$

$$0,35_{10} = 0,263_8$$

$$\begin{array}{r} \times 0,35 \\ \hline 16 \\ \times 5,60 \\ \hline 16 \\ \hline 9,60 \end{array}$$

$$0,35_{10} = 0,59_{16}$$



## Перевод чисел в десятичную систему счисления

*Пример.* Перевести число 1011,1 из двоичной системы счисления в десятичную.

разряды	3	2	1	0	-1	
число	1	0	1	1	,	$1_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11,5_{10}$

*Пример.* Перевести число 276,8 из восьмеричной системы счисления в десятичную.

разряды	2	1	0	-1	
число	2	7	6	,	$5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190,625_{10}$

*Пример.* Перевести число 1F3 из шестнадцатеричной системы счисления в десятичную.

разряды	2	1	0	
число	1	F	3	$_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$



## Перевод из восьмеричной и шестнадцатеричной системы счисления в двоичную

Заменить каждую цифру восьмеричного/шестнадцатеричного числа соответствующим трехразрядным/четыре разрядным двоичным кодом.

*Пример.* Перевести число  $527,1_8$  в двоичную систему счисления.

$$527,1_8 = \underbrace{101}_5 \underbrace{010}_2 \underbrace{111}_7, \underbrace{001}_1_2$$

*Пример.* Перевести число  $1A3,F_{16}$  в двоичную систему счисления.

$$1A3,F_{16} = \underbrace{0001}_1 \underbrace{1010}_A \underbrace{0011}_3, \underbrace{1111}_F_2$$



## Перевод из двоичной системы счисления в восьмеричную и шестнадцатеричную

Для перехода от двоичной к восьмеричной/шестнадцатеричной системе счисления поступают следующим образом: двигаясь от запятой влево и вправо, разбивают двоичное число на группы по 3(4) разряда, дополняя, при необходимости, нулями крайние левую и правую группы. Затем каждую группу из 3(4) разрядов заменяют соответствующей восьмеричной/шестнадцатеричной цифрой.

*Пример:*

$$010101001,101110_2 = 251,65_8$$



$$10101001,10111000_2 = A9,B8_{16}$$





## Перевод из восьмеричной системы счисления в шестнадцатеричную и обратно

При переходе из восьмеричной системы счисления в шестнадцатеричную и обратно вначале производится перевод чисел из исходной системы счисления в двоичную, а затем – в конечную систему .

*Пример.* Перевести число  $527,1_8$  в шестнадцатеричную систему счисления.

$$527,1_8 = \underbrace{000}_1 \underbrace{101010111}_5 \underbrace{,011}_7 \underbrace{0}_6 \underbrace{0}_2 = 157,6_{16}$$

*Пример.* Перевести число  $1A3,F_{16}$  в восьмеричную систему счисления.

$$1A3,F_{16} = \underbrace{110100011}_6 \underbrace{,1111}_4 \underbrace{00}_2 = 643,74_8$$





## Арифметические операции в позиционных системах счисления

Правила выполнения основных арифметических операций в любой позиционной системе счисления **подчиняются тем же законам**, что и в десятичной системе.

При **сложении** цифры суммируются по разрядам, и если при этом возникает переполнение разряда, то производится перенос в старший разряд. Переполнение разряда наступает тогда, когда величина числа в нем становится равной или большей основания системы счисления.

При **вычитании** из меньшей цифры большей в старшем разряде занимается единица, которая при переходе в младший разряд будет равна основанию системы счисления



Если при **умножении** однозначных чисел возникает переполнение разряда, то в старший разряд переносится число кратное основанию системы счисления. При умножении многозначных чисел в различных позиционных системах применяется алгоритм перемножения чисел в столбик, но при этом результаты умножения и сложения записываются с учетом основания системы счисления.

**Деление** в любой позиционной системе производится по тем же правилам, как и деление углом в десятичной системе, то есть сводится к операциям умножения и вычитания.

*Правила выполнения арифметических действий задаются таблицей:*

сложение	вычитание	умножение
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 * 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 * 0 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 * 1 = 1$



# Сложение в позиционных системах счисления

Цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево

двоичная система

$$\begin{array}{r}
 1\ 1\ 1 \\
 1\ 0\ 1\ 0\ 1 \\
 +\ 1\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 0\ 0\ 1\ 0
 \end{array}$$

$1+1=2=2+0$   
 $1+0+0=1$   
 $1+1=2=2+0$   
 $1+1+0=2=2+0$   
 $1+1=2=2+0$

Ответ:  $100010_2$

восьмеричная система

$$\begin{array}{r}
 1\ 1\ 1 \\
 2\ 1\ 5\ 4 \\
 +\ 7\ 3\ 6 \\
 \hline
 3\ 1\ 12
 \end{array}$$

$4+6=10=8+2$   
 $5+3+1=9=8+1$   
 $1+7+1=9=8+1$   
 $1+2=3$

Ответ:  $3112_8$

шестнадцатеричная система

$$\begin{array}{r}
 1\ 1 \\
 8\ D\ 8 \\
 +\ 3\ B\ C \\
 \hline
 C\ 9\ 4
 \end{array}$$

$8+12=20=16+4$   
 $13+11+1=25=16+9$   
 $8+3+1=12=C_{16}$

Ответ:  $C94_{16}$



## Вычитание в позиционных системах счисления

При вычитании чисел, если цифра уменьшаемого меньше цифры вычитаемого, то из старшего разряда занимается единица основания

двоичная  
система

$$\begin{array}{r}
 \overset{1}{1} \overset{1}{0} 1 0 1 \\
 - \quad 1 0 1 1 \\
 \hline
 0 1 0 1 0
 \end{array}$$

$1-1=0$   
 $2-1=1$   
 $0-0=0$   
 $2-1=1$

Ответ:  $1010_2$

восьмеричная  
система

$$\begin{array}{r}
 \overset{1}{4} \overset{1}{3} 5 0 6 \\
 - \quad 5 0 4 2 \\
 \hline
 3 6 4 4 4
 \end{array}$$

$6-2=4$   
 $8-4=4$   
 $4-0=4$   
 $8+3-5=11-5=6$

Ответ:  $36444_8$

шестнадцатеричная  
система

$$\begin{array}{r}
 \overset{1}{C} \overset{1}{9} 4 \\
 - \quad 3 B C \\
 \hline
 8 4 8
 \end{array}$$

$16+4-12=20-12=8$   
 $16+8-11=24-11=13=D_{16}$   
 $11-3=8$

Ответ:  $848_{16}$



# Умножение в позиционных системах счисления

При умножении многозначных чисел в различных позиционных системах применяется алгоритм перемножения чисел в столбик, но при этом результаты умножения и сложения записываются с учетом основания системы счисления

двоичная система

$$\begin{array}{r}
 \phantom{x} 11011 \\
 \phantom{x} \phantom{0} 1101 \\
 \hline
 11111011 \\
 111011 \\
 11011 \\
 \hline
 101011111
 \end{array}$$

$1+1+1=3=2+1$   
 $1+1+1=3=2+1$   
 $1+1=2=2+0$

Ответ:  
101011111

восьмеричная система

$$\begin{array}{r}
 \phantom{x} 163 \\
 \phantom{x} \phantom{0} 63 \\
 \hline
 1262 \\
 13351 \\
 \hline
 13351
 \end{array}$$

$3 \cdot 3 = 9 = 8 + 1$   
 $6 \cdot 3 + 1 = 19 = 16 + 3 = 2 \cdot 8 + 3$   
 $1 \cdot 3 + 2 = 5$   
 $6 \cdot 3 = 18 = 16 + 2 = 8 \cdot 2 + 2$   
 $6 \cdot 6 + 2 = 38 = 32 + 6 = 4 \cdot 8 + 6$   
 $6 \cdot 1 + 4 = 10 = 8 + 2$   
 $6 + 5 = 11 = 8 + 3$

Ответ: 13351<sub>8</sub>

# Умножение в шестнадцатеричной системе счисления

Выполните умножение чисел  $20A4_{16} * B15_{16}$

$$\begin{array}{r} \times \quad 20A4_{16} \\ \quad B15_{16} \\ \hline \end{array}$$

$$\begin{aligned} 4 * 5 &= 20 = 1 * 16 + 4 \\ A_{16} * 5 + 1 &= 10_{10} * 5 + 1 = 51 = 3 * 16 + 3 \\ 0 * 5 + 3 &= 3 \\ 5 * 2 &= 10 = A_{16} \end{aligned}$$

$$A334$$

$$20A4$$

$$1670C$$

$$\begin{aligned} B_{16} * 4 &= 11_{10} * 4 = 44 = 2 * 16 + 12_{10} = 2 * 16 + C_{16} \\ B_{16} * A_{16} + 2 &= 11_{10} * 10_{10} + 2 = 112 = 7 * 16 + 0 \\ B_{16} * 0 + 7 &= 7 \\ B_{16} * 2 &= 11 * 2 = 22 = 1 * 16 + 6 \end{aligned}$$

$$169B974_{16}$$

$$3 + A_{16} + C_{16} = 3 + 10_{10} + 12_{10} = 25 = 1 * 16 + 9$$

$$A_{16} + 0 + 0 + 1 = 10 + 1 = 11_{10} = B_{16}$$

ОТВЕТ:  $169B974_{16}$



## Деление в позиционных системах счисления

Деление в любой позиционной системе производится по тем же правилам, как и деление углом в десятичной системе. При этом необходимо учитывать основание системы счисления.

двоичная  
система

$$\begin{array}{r|l} 100011 & 1110 \\ - 1110 & \hline \hline \phantom{1}1110 & 10,1 \\ - \phantom{1}1110 & \\ \hline \phantom{1}0 & \end{array}$$

Ответ:  $10,1_2$

восьмеричная  
система

$$\begin{array}{r|l} 13351 & 163 \\ - 1262 & \hline \hline \phantom{1}531 & 63 \\ - \phantom{1}531 & \\ \hline \phantom{1}0 & \end{array}$$

Ответ:  $63_8$

# Представление чисел в компьютере



формат с фиксированной запятой – целые числа  
формат с плавающей запятой – вещественные числа

**Вещественные числа** хранятся и обрабатываются в компьютере в формате с плавающей запятой. Этот формат базируется на экспоненциальной форме записи, в которой может быть записано любое число.

Целые числа в компьютере могут представляться **со знаком или без знака**.

**Целые числа без знака** занимают в памяти один или два байта.

Формат числа в байтах	Запись с порядком	Обычная запись
1	$0 \dots 2^8 - 1$	0 ... 255
2	$0 \dots 2^{16} - 1$	0 ... 65535

*Пример.* Число  $72_{10} = 1001000_2$  в однобайтовом формате

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





**Целые числа со знаком** занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.

Знак «плюс» кодируется нулем, а «минус» - единицей

Формат числа в байтах	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7 - 1$	-128 ... 127
2	$-2^{15} \dots 2^{15} - 1$	-32 768 ... 32 767
4	$-2^{31} \dots 2^{31} - 1$	-2 147 483 648 ... 2 147 483 647

*Пример.* Число  $-72_{10} = -1001000_2$  в однобайтовом формате

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

*Пример.* Число  $62_{10} = 111110_2$  в однобайтовом формате

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



В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком:

**прямой** код, **обратный** код и **дополнительный** код.

**Прямой код** – чаще всего отводится 2 байта памяти (16 бит), в старший разряд записывается «0» если число положительное и «1» – если число отрицательное.

**Обратный код** – для положительных чисел совпадает с прямым кодом, для отрицательных чисел образуется из прямого кода заменой нулей единицами, а единиц – нулями, кроме цифр знакового разряда.

**Пример.** Число  $-57_{10} = -111001_2$

прямой	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1
обратный	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0



**Дополнительный код** используется для представления отрицательных чисел, позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие.

Дополнительный код отрицательного числа  $A$ , хранящегося в  $n$  ячейках равен  $2^n - |A|$ . Образуется из обратного кода с последующим прибавлением единицы к его младшему разряду

	$13_{10}$	$-12_{10}$
Прямой код	00001101	10001100
Обратный код	—	11110011
Дополнительный код	—	$\begin{array}{r} 11110011 \\ + \quad \quad 1 \\ \hline 11110100 \end{array}$



Число	- 128	- 127	- 0
<b>Прямой код модуля</b>	1000 0000	0111 1111	0000 0000
<b>Обратный код</b>	0111 1111	1000 0000	1111 1111
<b>Дополнительный код</b>	1000 0000	1000 0001	0000 0000

Отрицательные десятичные числа при вводе в компьютер автоматически преобразуются в обратный или дополнительный код и в таком виде хранятся, перемещаются и участвуют в операциях.

При выводе таких чисел из компьютера происходит обратное преобразование в отрицательные десятичные числа.



Определим диапазон чисел, которые могут храниться в оперативной памяти в формате *длинных целых чисел со знаком* (отводится 32 бита памяти): минимальное число  $-2^{31} = -2147483648$

максимальное число  $2^{31} - 1 = 2147483647$

Разрядность	8	16	32
Минимум (без знака)	0	0	0
Максимум (без знака)	255	65535	4294967295
Минимум (со знаком)	-128	-32768	-2147483648
Максимум (со знаком)	127	32767	2147483647

😊 – простота и наглядность представления чисел, простота алгоритмов реализации арифметических операций

😞 – небольшой диапазон представляемых чисел, недостаточный для решения большинства прикладных задач



# Представление вещественных чисел в компьютере

Любое число  $N$  в системе счисления с основанием  $q$  можно записать в виде  $N = m \cdot q^p$ , где  $m$  называется **мантиссой** числа, а  $p$  – **порядком**.

Такой способ записи чисел называется **представлением числа с плавающей точкой**.

## Характеристики форматов вещественных чисел

Формат числа	Диапазон абсолютных значений	Биты мантиссы	Размер в байтах (битах)
одинарный вещественный	$10^{-45} \dots 10^{38}$	23+1	4 (32)
двойной	$10^{-39} \dots 10^{38}$	используется редко	6 (48)
расширенный	$10^{-324} \dots 10^{308}$	52+1	8 (64)
	$10^{-4932} \dots 10^{4932}$	64+1	10 (80)



Способ представления числа с плавающей запятой, опирается на **нормализованную (экспоненциальную)** запись действительных чисел.

**Нормализованной** называется запись отличного от нуля действительного числа в виде  $m \cdot r^q$ , где  $q$  – целое число (положительное, отрицательное или ноль), а  $m$  – правильная  $P$ -ичная дробь, у которой первая цифра после запятой не равна нулю. Мантисса нормализованного числа может изменяться в диапазоне:  $1/q \leq |m| < 1$ .

В нормализованных числах цифра после точки всегда должна быть значащей.

Пример:  $0,0836 \cdot 10^3 = 0,836 \cdot 10^2$   
ненормализованное число      нормализованное число

При этом, часть разрядов отводится для записи порядка числа, а остальные разряды – для записи мантиссы. По одному разряду в каждой группе отводится для изображения знака порядка и знака мантиссы (0 – если знак плюс, 1 – знак минус).

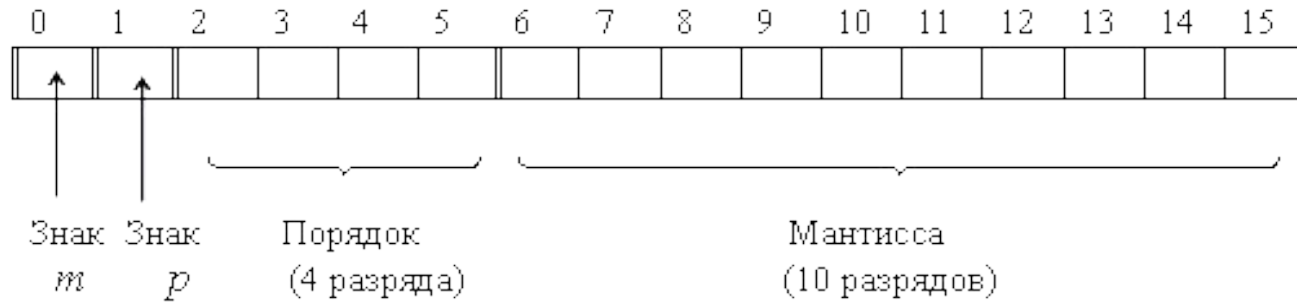
$s_p$	$s_q$	$b_k$	$b_{k-1}$	...	$b_2$	$b_1$	$a_1$	$a_2$	...	$a_{n-1}$	$a_n$
-------	-------	-------	-----------	-----	-------	-------	-------	-------	-----	-----------	-------

Следующие  $k$ -разрядов используются для изображения абсолютной величины порядка числа, остальные  $n$ -разрядов используются для изображения абсолютной величины мантиссы.

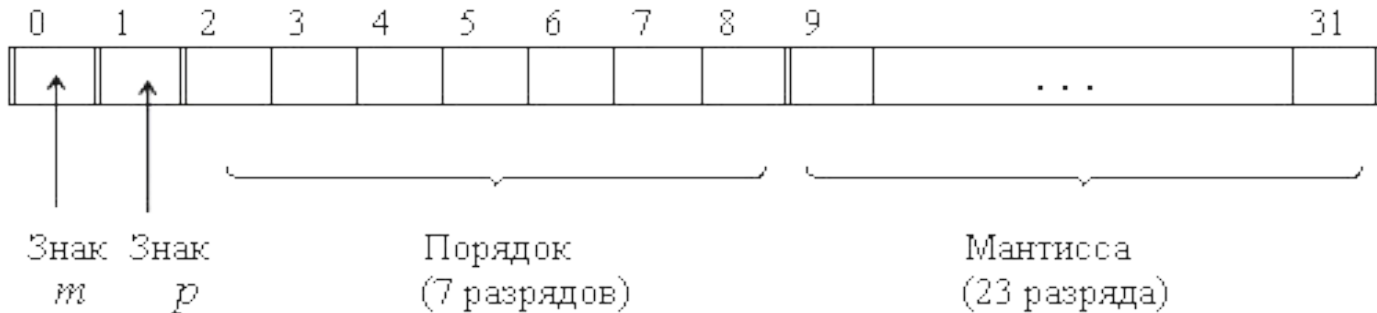


Порядок и мантисса определяют диапазон изменения чисел и их точность.

Представление чисел в порядке полуслова:



Представление чисел в порядке слова:



*Примеры:*

$$3,1415926 = 0,31415926 * 10^1;$$

$$1000 = 0,1 * 10^4;$$

$$\mathbf{0,123456789 = 0,123456789 * 10^0};$$

$$0,0000107_8 = 0,1078 * 8^{-4}; \text{ (порядок записан в 10-й системе)}$$

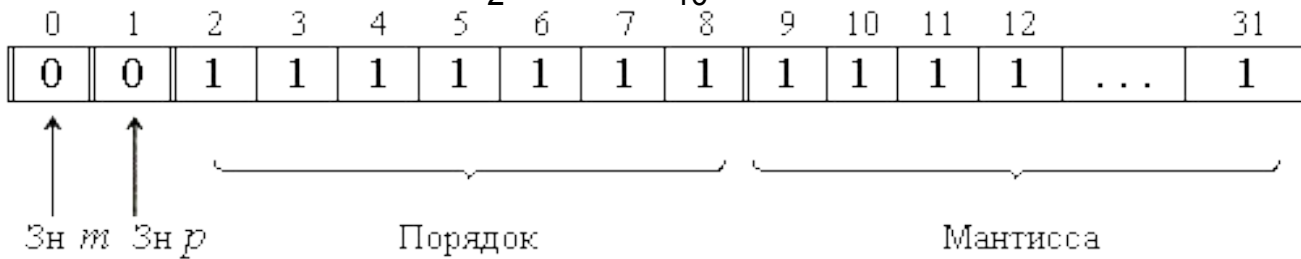
$$1000,0001_2 = 0,100000012 * 2^4.$$





Максимальным числом представимым в формате слова будет

$$A = (0.1111\dots 1 \cdot 10^{11111111})_2 (1 \cdot 2^{127})_{10}$$



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. Точность может быть повышена путем увеличения количества разрядов мантиссы. Это реализуется путем представления чисел с так называемой двойной точностью (используется формат двойного слова).

При записи основания числа в десятичной системе можно говорить о нормализованной записи: мантиссу и порядок  $q$ -ичного числа записывают в системе счисления с основанием  $q$ .

Различают:

- научная нормализованная запись числа:  $1 \leq |m| < 10, q = 10, (3,5 \cdot 10^2)$
- инженерная нормализованная запись (информатика):  

$$0,1 < |m| \leq 1, q = 10, (0,35 \cdot 10^2)$$
- компьютерная нормализованная запись:  $1 \leq |m| < 10, q = 10(E), (3,5E2)$



## Выполнение арифметических операций над числами с плавающей запятой

**При сложении и вычитании чисел** – производится **выравнивание порядка**. После этой операции одинаковые разряды чисел оказываются расположенными в одних и тех же по номеру разрядах ячеек, отведенных под операнды, и теперь уже сложение или вычитание мантисс выполняется достаточно просто, так же, как над целыми числами.

$$10^2 * 0,23619$$

---

$$10^{-2} * 0,71824, \text{ меньший порядок «приводится» к большему}$$

$$10^2 * 0,23619$$

$$10^2 * 0,00071824$$

---

$$10^2 * 0,23690824$$

**При умножении** двух целых чисел с плавающей запятой их порядки необходимо просто сложить, а мантиссы – перемножить без предварительного выравнивания.

**При делении** из порядка делимого надо вычесть порядок делителя, а мантиссу делимого разделить на мантиссу делителя.



В 1976 г. была принята инициатива создать **единый стандарт для представления чисел с плавающей запятой.**

Свои предложения по единому стандарту представили компании DEC, National Superconductor, Zilog, Motorola.

### VAX от DEC

Спецификация VAX была значительно проще, уже была реализована в компьютерах PDP-11, и было件нятно, как на ней получить максимальную производительность.

### «К-С-S» от Intel

С другой стороны в «К-С-S» содержалось много полезной функциональности, такой как «специальные» и «денормализованные» числа. В «К-С-S» все арифметические алгоритмы заданы строго. Строгость арифметики «К-С-S» сделала возможным доказательство теорем, опираясь на арифметику с плавающей запятой.

Разработчики «К-С-S» победили и их детище воплотилось в стандарт IEEE754. Числа с плавающей запятой в нем представлены в виде знака (s), мантиссы (M) и порядка (E) следующим образом:

$$(-1)^s \times 1.M \times 2^E$$



# IEEE 754 - стандарт двоичной арифметики с плавающей точкой

Данный стандарт разработан ассоциацией IEEE (*Institute of Electrical and Electronics Engineers*) и используется для представления действительных чисел (чисел с плавающей точкой) в двоичном коде.

Наиболее используемый стандарт для вычислений с плавающей точкой, используется многими микропроцессорами и логическими устройствами, а также программными средствами.

Полное название стандарта в ассоциации IEEE:

□ *IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-1985)*  
*IEEE стандарт для двоичной арифметики с плавающей точкой (ANSI/IEEE Std 754-1985)*

Название стандарта в международной электротехнической комиссии IEC:

□ *IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems*  
*IEC 60559:1989 двоичная арифметика с плавающей точкой для микропроцессорных систем*

Стандарт содержит 23 страницы текста в 7 секциях и одном приложении.



## Специальные числа: ноль, бесконечность и неопределенность

В IEEE754 **число «0»** представляется значением с порядком, равным  $E=E_{\min}-1$  (для single это -127) и нулевой мантиссой.

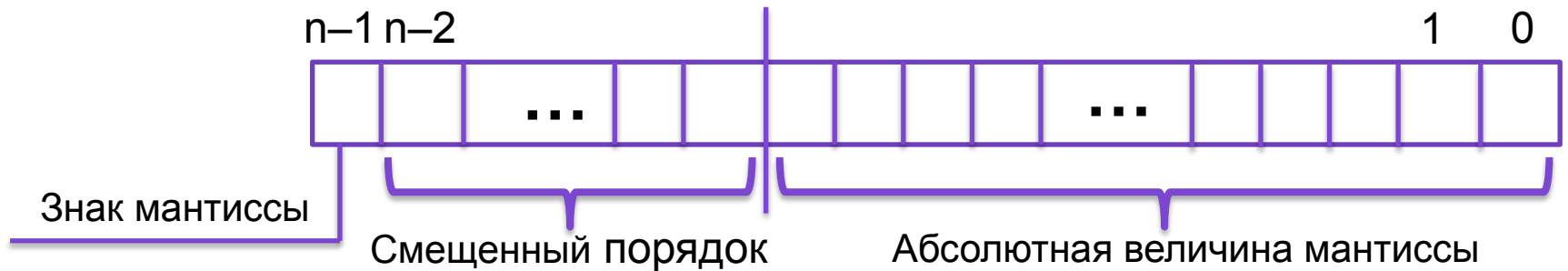
**Бесконечности** представлены как числа с порядком  $E=E_{\max}+1$  и нулевой мантиссой. Получить бесконечность можно при переполнении и при делении ненулевого числа на ноль. Бесконечность при делении определили исходя из существования пределов, когда делимое и делитель стремятся к какому-то числу. Соответственно,  $c/0=\pm\infty$  (например,  $3/0=+\infty$ , а  $-3/0=-\infty$ ), так как если делимое стремиться к константе, а делитель к нулю, предел равен бесконечности.

При  $0/0$  предел не существует, поэтому результатом будет неопределенность.

**Неопределенность** или NaN (от not a number) – это представление, придуманное для того, чтобы арифметическая операция могла всегда вернуть какое-то не бессмысленное значение. NaN представлен как число, в котором  $E=E_{\max}+1$ , а мантисса не нулевая. Любая операция с NaN возвращает NaN.



Все компьютеры поддерживают несколько международных стандартных форматов, различающихся по точности, но имеющих одинаковую структуру следующего вида:



Здесь порядок  $n$ -разрядного нормализованного числа задается в **смещенной форме**: если для задания порядка выделено  $k$  разрядов, то к истинному значению порядка, **представленного в дополнительном коде**, прибавляют смещение, равное  $(2^{k-1} - 1)$ .

Например, порядок, принимающий значения в диапазоне от  $-128$  до  $+127$ , представляется смещенным порядком, значения которого меняются от  $0$  до  $255$ .

**Смещенный порядок** позволяет не хранить знак порядка.

**Смещенный порядок** рассчитывается по формуле  $2^{a-1} + \text{ИП}$ ,

где  $a$  – число разрядов, отводимых под порядок (выделенных для представления порядка числа в формате с плавающей запятой),

**ИП** – (истинный) порядок числа.

*Пример:* если истинный порядок равен  $-5$ , тогда смещенный порядок для 4-байтового числа будет равен  $127 - 5 = 122$ .



В компьютере, число с плавающей запятой представляется в виде набора отдельных двоичных разрядов, условно разделенных на **знак, порядок и мантиссу**.

Знак (0 или 1)	Порядок + смещение	Мантисса ( <i>m</i> )
-------------------	-----------------------	--------------------------



Прибавление смещения позволяет записывать положительные и отрицательные порядки в виде положительных чисел.

Рассмотрим пример записи числа с плавающей точкой:

- ▶ число +178.25
- ▶ в двоичной системе счисления  $+1011\ 0010.01 = +1.0110\ 0100\ 1 \times 2^{111}$

<b>Одинарная точность (32 бита, смещение порядка <math>127_{10} = 1111111_2</math>)</b>		
Знак	1 бит	0
Порядок	8 бит	<b>111</b> +111 1111= <b>1000 0110</b>
Мантисса	23 бита	011 0010 0100 0000 0000 0000
Итого	32 бита	<b>0100 0011 0</b> 011 0010 0100 0000 0000 0000
В 16-ричном формате		<b>43324000</b>



## Стандартные форматы представления вещественных чисел:

Тип данных	Длина (бит)			Смещение порядка
	число	порядок	мантисса	
одинарная точность	32	8	23	127
двойная точность	64	11	52	1023
расширенная точность	80	15	64	16383

Вещественный формат с  $m$ -разрядной мантиссой позволяет абсолютно точно представлять  $m$ -разрядные целые числа, т. е. **любое двоичное целое число, содержащее не более  $m$  разрядов, может быть без искажений преобразовано в вещественный формат**

В 2008 года ассоциация IEEE выпустила стандарт IEEE 754-2008, который включил в себя стандарт IEEE 754-1985.

В новом стандарте IEEE 754-2008 кроме чисел с основанием 2 присутствуют числа с основанием 10, так называемые *десятичные (decimal)* числа с плавающей запятой.





## *Алгоритм представления числа с плавающей запятой*

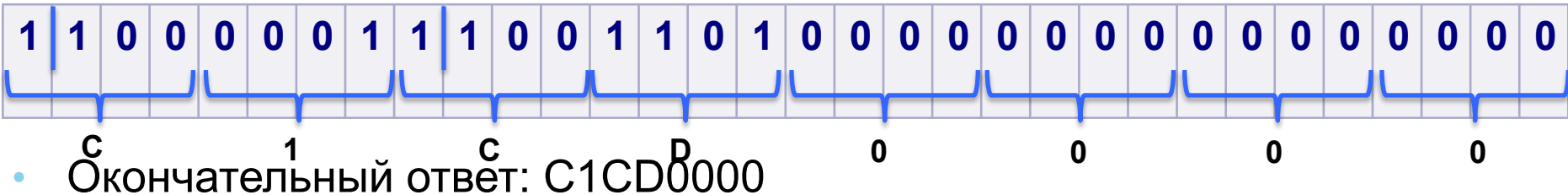
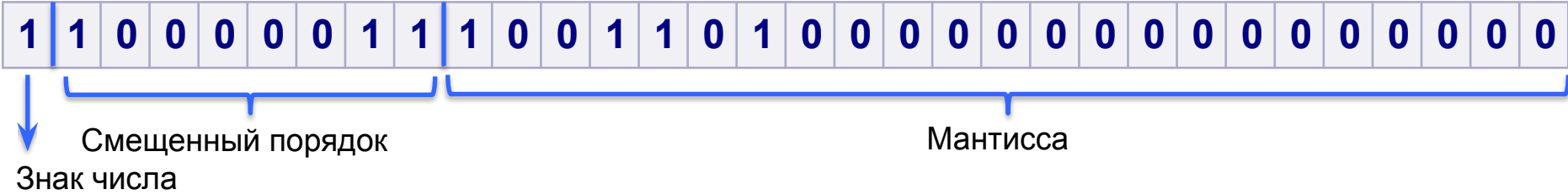
- Перевести число из  $r$ -ичной системы счисления в двоичную;
- представить двоичное число в нормализованной экспоненциальной форме;
- рассчитать смещённый порядок числа;
- разместить знак, порядок и мантиссу в соответствующие разряды сетки.



# Пример

- Представить число  $-25,625$  в машинном виде с использованием 4 байтового представления (где 1 бит отводится под знак числа, 8 бит - под смещённый порядок, остальные биты - под мантиссу).

- 1.  $25_{10} = 11001_2$   
 $0,625_{10} = 0,101_2$   
 $-25,625_{10} = -11001,101_2$
- 2.  $-11001,101_2 = -1,1001101_2 * 2^4$
- 3. СП=127+4=131
- 4.



- Окончательный ответ: C1CD0000



Информация почти всегда является ответом.

Наиболее простые вопросы те, на которые можно дать ответы «ДА» или «НЕТ».

В кибернетике и информатике за единицу информации принято считать такое количество информации, при котором из двух равновероятных возможностей можно выбрать одну. Такая единица информации называется БИТ.

Количество информации, которое необходимо для получения ответа при выборе из нескольких возможностей, равно логарифму по основанию 2 от числа возможностей. Один бит информации равен  $\text{Log}_2 2 = 1$ .

Единицы измерения объема информации:

1 байт = 8 бит

1 килобайт = 1024 байт

1 мегабайт = 1024 килобайта

1 гигабайт = 1024 мегабайта

1 терабайт = 1024 гигабайт =  $2^{40}$  байт (примерно  $10^{12}$  байт), иногда

называют *тонна*.

1 петабайт = 1024 терабайт =  $2^{50}$  байт (примерно  $10^{15}$  байт)

1 эксабайт = 1024 петабайт =  $2^{60}$  байт (примерно  $10^{18}$  байт)

1 зеттабайт = 1024 эксабайт =  $2^{70}$  байт (примерно  $10^{21}$  байт)

1 йоттабайт = 1024 зеттабайт =  $2^{80}$  байт (примерно  $10^{24}$  байт) и т.д.



# Кодирование текстовой информации

Соответствие между набором символов и набором числовых значений называется **кодировкой символа**. При вводе в компьютер текстовой информации происходит ее двоичное кодирование. Код символа хранится в оперативной памяти компьютера. В процессе вывода символа производится обратная операция – **декодирование**, т.е. преобразование символа в его изображение.

Институтом стандартизации США была введена в действие система кодирования **ASCII** (*American Standard Code for Information Interchange*). Каждому символу ASCII соответствует 8–битовый двоичный код

**(1 символ – 1 байт).**

В последнее время широкое распространение получил новый международный стандарт **Unicode**. Стандарт состоит из двух основных разделов: универсальный набор символов (*UCS, universal character set*) и семейство кодировок (*UTF, Unicode transformation format*).

Каждому символу Unicode соответствует 16–битовый двоичный код

**(1 символ – 2 байта).**

Dec	Hex	Oct	Char	Description
0	0	000		null
1	1	001		start of heading
2	2	002		start of text
3	3	003		end of text
4	4	004		end of transmission
5	5	005		enquiry
6	6	006		acknowledge
7	7	007		bell
8	8	010		backspace
9	9	011		horizontal tab
10	A	012		new line
11	B	013		vertical tab
12	C	014		new page
13	D	015		carriage return
14	E	016		shift out
15	F	017		shift in

Dec	Hex	Oct	Char
64	40	100	@
65	41	101	A
66	42	102	B
67	43	103	C
68	44	104	D
69	45	105	E
70	46	106	F
71	47	107	G
72	48	110	H
73	49	111	I
74	4A	112	J
75	4B	113	K
76	4C	114	L
77	4D	115	M
78	4E	116	N
79	4F	117	O

## Пример

С помощью кодировок ASCII и Unicode закодирована фраза:

«Я поступил в университет!».

Оцените информационный объем этой фразы.

*Решение.*

В данной фразе содержится 25 символов, включая пробелы и знак препинания.

В кодировке ASCII на 1 символ отводится 1 байт, следовательно для фразы

понадобится 25 байт или 200 бит.



С помощью кодировок ASCII и Unicode закодирована фраза:

«Я поступил в университет!».

Оцените информационный объем этой фразы.

В кодировке Unicode 1 символ занимает 2 байта, поэтому вся фраза займет 50 байт или 400 бит.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	Ў	Ғ	,	Ғ	„	...	†	‡	€	‰	Ө	Ҝ	Ң	Қ	Һ	Ү
90	ұ	‘	’	“	”	•	–	—	□	™	ө	ҝ	ң	қ	һ	ү
A0		У	ў	Җ	#	Ҳ	!	§	Ё	©	Є	«	¬	-	®	Ї
B0	°	±	І	і	ҫ	μ	¶	·	ё	№	є	»	җ	ө	ә	ї
C0	A	B	B	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	a	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я



# Формула Хартли

Для измерения количества информации, которое может быть передано при помощи алфавита, существует формула Хартли

$n = p^i$ , где  $n$  – число равновероятных событий,  $i$  – количество информации, полученной в результате совершения события,  $p$  – количество различных вариантов

или

$p$  – количество используемых символов,  $i$  – длина строки символов или сигналов.

*Пример:*

*Сколько различных сигналов можно записать с помощью 32-разрядного компьютерного кода.*

*В этом случае длина строки – 32, количество используемых символов – 2, следовательно  $n = 2^{32} = 4\ 294\ 967\ 296$ .*



# Кодирование графической информации

Растровая графика

[Empty rectangular box]

[Large empty rounded rectangular box]

Векторная графика

[Empty rectangular box]

[Large empty rounded rectangular box]





Мониторы могут работать в двух режимах: *текстовом и графическом.*

В **текстовом режиме** экран разбивается на 25 строк по 80 символов.

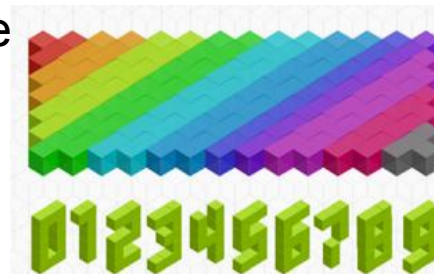
Каждая позиция экрана называется **знакоместом**, в нем находится один символ.

Символ может быть высвечен одним из 16 цветов, можно изменять и цвет фона (8 цветов). Для описания каждого знакоместа потребуется 2 байта: 1 байт – символ, 2 байт – его цветовые характеристики (цвет символа 4 бита, цвет фона 3 бита и 1 бит для представления мерцания (0 – не мерцает, 1 – мерцает).

Таким образом содержимое экрана занимает в видеопамяти компьютера  $25 \cdot 80 \cdot 2 = 4000$  байт = 4 Кбайта.

В **графическом режиме** экран разделяется по **пиксели**, количество которых определяет разрешающую способность монитора и зависит от его типа и режима.

Такая форма представления информации называется растровой. Каждый пиксель может иметь цвет (16, 256, 16000 и т.д.), 256 цветов – 1 байт. При объемном характере раstra ячейка называется **вокселем**: образовано из слов: объёмный (англ. *volumetric*) и пиксел (англ. *pixel*), содержит значение элемента раstra в трехмерном пространстве. Трёхмерная матрица хранит значение вокселя для каждого единичного элемента объёмного пространства, поэтому **воксельные модели** хорошо подходят для моделирования. Для воксельных моделей существует множество **алгоритмов визуализации.**

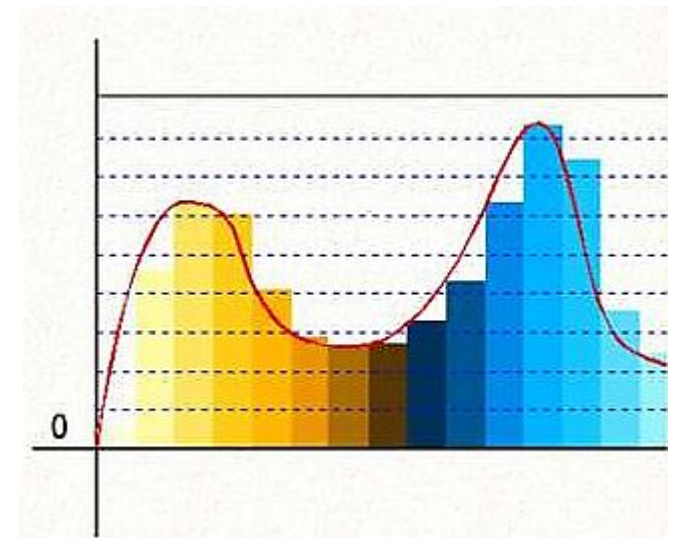
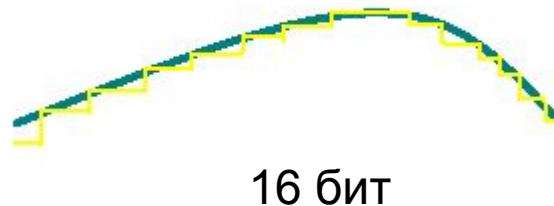




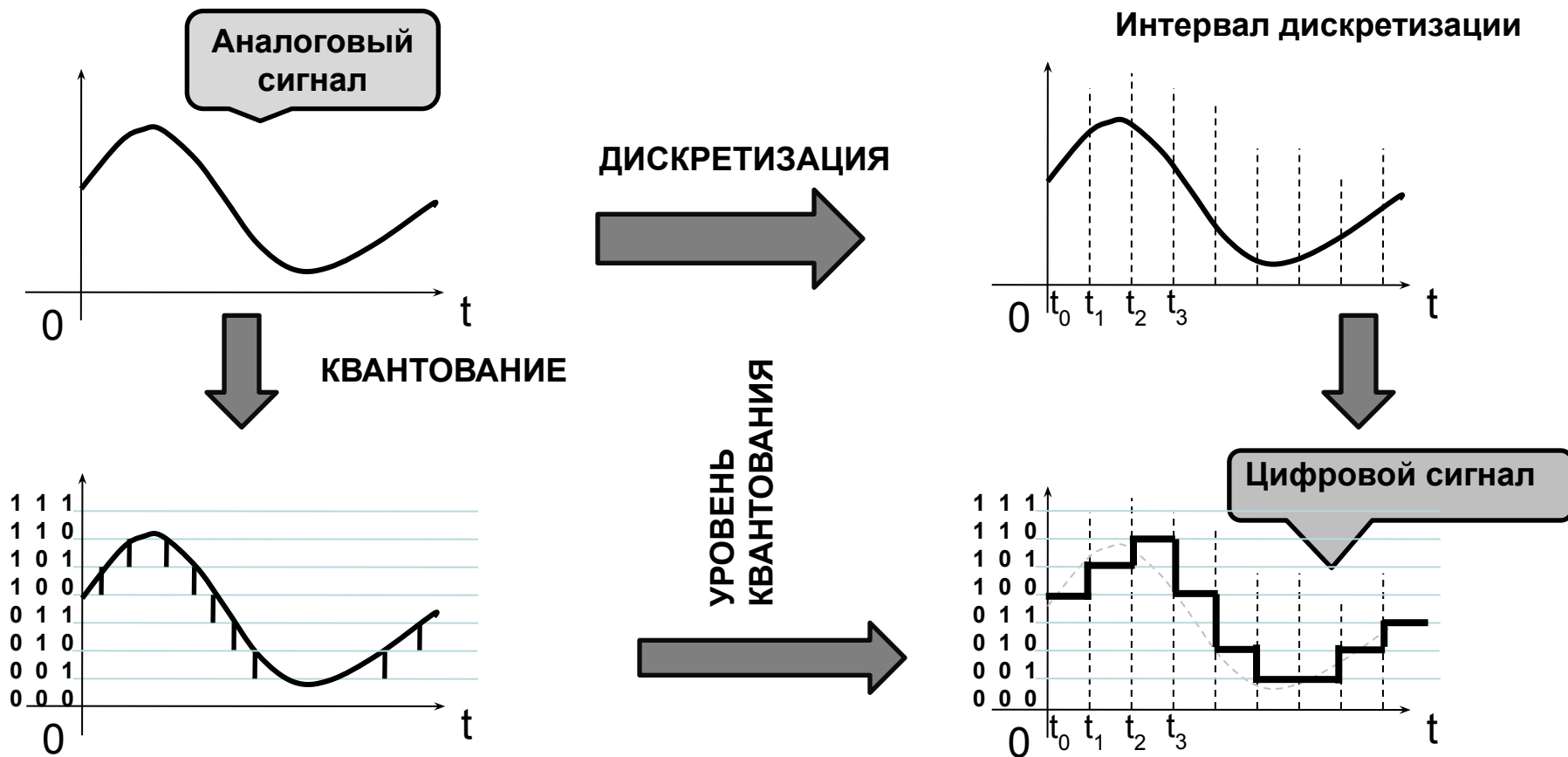
# Кодирование звуковой информации

Существует (1999 г.) два способа звукозаписи: **цифровая** запись (дискретизация – измерения количество раз в секунду) и **MIDI-запись** – запись определенных команд – указаний (электронный эквивалент нотной записи). Разработана в начале 80-х годов, для записи команд необходимо устройство, выполняющее роль клавишного синтезатора, которое воспринимает команды и при их получении генерирует соответствующие звуки.

Звуковая плата преобразует звук на входе в цифровую информацию путем измерения его характеристик (период, амплитуда). При воспроизведении звука специальное устройство звуковой платы преобразует цифры обратно в аналоговый сигнал.



# Представление сигнала в двоичной системе после аналого-цифрового преобразования



(100), (101), (110), (100), (010), (001), (001), (010), (011).....

Получили входные данные для обработки цифрового сигнала



## КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1) Перечислите типы систем счисления.
- 2) Что является основанием системы счисления, перечислите алфавит нескольких систем счисления.
- 3) Укажите представление числа в позиционной системе счисления.
- 4) Опишите алгоритм перевода целых чисел в другую систему счисления.
- 5) Опишите алгоритм перевода дробных чисел в другую систему счисления.
- 6) Перечислите и приведите примеры арифметических операций в позиционных системах счисления.
- 7) Укажите правила хранения целых и вещественных чисел в компьютере.
- 8) Что такое прямой, обратный и дополнительный код?
- 9) Что такое нормализованная запись числа?
- 10) Приведите алгоритм выполнения арифметических операций над числами с плавающей запятой.
- 11) Перечислите стандартные форматы представления вещественных чисел.
- 12) Приведите пример перевода числа с плавающей запятой в машинный формат.
- 13) Перечислите виды кодирования текстовой информации.
- 14) Опишите виды кодирования графической информации в компьютере.
- 15) Укажите алгоритм кодирования звуковой информации в компьютере.

*Спасибо за внимание !!!*