



**Кафедра «Автоматизированные станочные системы»**  
**Dept. of Automated Manufacturing Systems**

---

# **Системы счисления, или Как считает компьютер?**

# Системы счисления

**Система счисления (numbering system)** - совокупность приемов и правил для записи чисел знаками.

Способов записи чисел цифровыми знаками существует бесчисленное множество.

Наиболее известна десятичная система счисления, в которой для записи чисел используются цифры 0, 1, ..., 9.

Любая предназначенная для практического применения система счисления должна обеспечивать:

- возможность представления любого числа в рассматриваемом диапазоне величин;
- единственность представления (каждой комбинации символов должна соответствовать одна и только одна величина);
- простоту оперирования числами

# Непозиционные системы счисления

Непозиционная система счисления – система, для которой **численное значение символа (цифры) не зависит от его положения в числе.**

Например, система с одним символом-палочкой встречалась у многих народов. Для изображения какого-то числа в этой системе нужно записать количество палочек, равное данному числу.

$$2=|| \quad 5=||||| \quad 10=||||||||| \quad 1250=|||||||||.....$$

Эта система неэффективна, так как запись числа получается длинной.

Основной недостаток всех непозиционных систем – громоздкость записи и трудность выполнения арифметических действий

$$||||||||| \times ||||||||||| = ?$$

## Самая известная «почти» непозиционная система счисления - римская



Римская система использует набор  
следующих символов (цифр):

I - "один";

V - "пять";

X - "десять";

L - "пятьдесят";

C - "сто";

D - "пятьсот";

M - "тысяча"

В этой системе существует **отклонение от правила независимости значения цифры от положения в числе**. В числах LX и XL символ X принимает два различных значения: +10 – в первом случае и –10 – во втором случае.

Запись больших чисел громоздка: **1994=MDCCLXXXIV**

Выполнение  
арифметических операций  
крайне затруднено:



XXXVII

IX

—————  
???????

# Позиционные системы счисления

Позиционная система счисления – система, в которой **значение символа (цифры) определяется его положением в числе**: один и тот же знак принимает различное значение. Например, в десятичном числе 222 первая цифра справа означает **две единицы**, соседняя с ней – **два десятка**, а левая – **две сотни**.

Любая позиционная система характеризуется **основанием**. Основание (базис) позиционной системы счисления – количество знаков или символов, используемых для изображения числа в данной системе.

Позиционные системы счисления имеют ряд преимуществ перед непозиционными: удобство выполнения арифметических и логических операций, а также представление больших чисел, поэтому и людьми, и в цифровой технике применяются позиционные системы счисления.

# Различные используемые позиционные системы счисления

В общем виде для любой позиционной системы счисления с основанием  $B$  число из  $n$  цифр имеет вид:

$$A_B = a_1 B^{n-1} + a_2 B^{n-2} + \dots + a_n B^0$$

Например:

$$123_{10} = 1 \cdot 10^{3-1} + 2 \cdot 10^{3-2} + 3 \cdot 10^{3-3}$$

$B$	Название	Цифры
2	двоичная	0, 1
3	троичная	0, 1, 2
8	восьмеричная	0, ..., 7
16	шестнадцатеричная	0, ..., 9, A, ..., F

Каждая система счисления имеет свои правила арифметики (таблицы умножения, сложения). Поэтому, производя какие-либо операции над числами, надо помнить о системе счисления, в которой они представлены.

# Если основание больше 10...

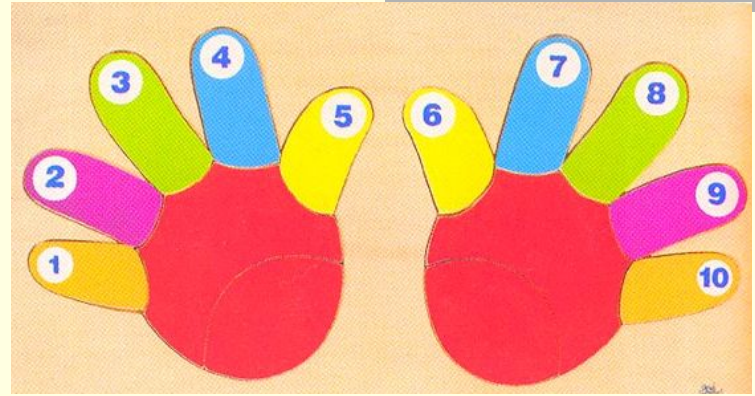
Если основание системы  $q$  превышает 10, то цифры, начиная с 10, при записи обозначают прописными буквами латинского алфавита: A, B, ..., Z. При этом цифре 10 соответствует знак 'A', цифре 11 - знак 'B' и т.д.

В таблице приведены десятичные числа от 0 до 15 и их эквивалент в различных системах счисления.

B=10	B=2	B=16	B=10	B=2	B=16
0	0	0	8	1000	8
1	1	1	9	1001	9
2	10	2	10	1010	A
3	11	3	11	1011	B
4	100	4	12	1100	C
5	101	5	13	1101	D
6	110	6	14	1110	E
7	111	7	15	1111	F

# Применение различных систем счисления

**Десятичная** (число пальцев на руках) – исторически стала единственной системой, применяемой населением Земли.



В **двоичной системе** работает электронная техника, так как всего две цифры 0 и 1 легче всего представить в виде электрических сигналов

В **шестнадцатеричной системе** представляются многие числа в Интернете (скажем, так кодируется цвет в HTML):  
`background=#ffffff;`



# Общий метод перевода чисел из одной СС в другую

Перевод целых чисел из системы с основанием  $q_1$  в систему с основанием  $q_2$  осуществляется делением на основание  $q_2$  новой системы счисления, правильных дробей – умножением на основание  $q_2$ . Действия деления и умножения выполняются по правилам  $q_1$ -арифметики. Перевод дробей осуществляется отдельно по указанным правилам, результат записывается в виде новой дроби в системе с основанием  $q_2$ .

**Пример.** Перевести десятичное число  $A = 61_{10}$  ( $q_1 = 10$ ) в двоичную систему счисления ( $q_2 = 2$ ).

$$\begin{array}{r} \underline{61} \quad | \underline{2} \\ \underline{60} \quad - \quad 30 \quad | \underline{2} \\ b_0 = 1 \quad \underline{30} \quad - \quad 15 \quad | \underline{2} \\ b_1 = 0 \quad \underline{14} \quad - \quad 7 \quad | \underline{2} \\ b_2 = 1 \quad \underline{6} \quad - \quad 3 \quad | \underline{2} \\ b_3 = 1 \quad \underline{2} \quad - \quad 1 \quad | \underline{2} \\ b_4 = 1 \quad \underline{1} \end{array}$$

Полученные остатки от деления записываем по порядку появления в ряд **справа налево**

Получаем правильный ответ:  $61_{10} = 111101_2$

# Табличный метод перевода

В простейшем виде табличный метод заключается в следующем: имеется таблица всех чисел одной системы с соответствующими эквивалентами из другой системы; задача перевода сводится к нахождению соответствующей строки таблицы и выбору из нее эквивалента. Такая таблица очень громоздка и требует большой емкости памяти для хранения.

Другой вид табличного метода заключается в том, что имеются таблицы эквивалентов в каждой системе только для цифр этих систем и степеней основания (положительных и отрицательных); задача перевода сводится к тому, что в выражение ряда для исходной системы счисления надо поставить эквиваленты из новой системы для всех цифр и степеней основания и произвести соответствующие действия (умножения и сложения) по правилам  $q_2$ -арифметики. полученный результат этих действий будет изображать число в новой системе счисления.

# Пример табличного метода перевода

**Пример.** Перевести десятичное число  $A = 113_{10}$  в двоичную систему счисления, используя таблицу эквивалентов цифр и степеней основания.

Десятичное число	Двоичное число
$10^0$	0001
$10^1$	1010
$10^2$	110 0100

**Решение.** Подставив значения двоичных эквивалентов десятичных цифр и степеней основания, получим:

$$A = 113_{10} = 1 \cdot 10^2 + 1 \cdot 10^1 + 3 \cdot 10^0 = 0001 \cdot 1100100 + 0001 \cdot 1010 + 0011 \cdot 0001 = 1110001_2.$$

**Ответ:**  $1110001_2$ .

# Какие бывают числа

**Целые:** 10, 125, -1512

**Действительные** (вещественные) – с дробной частью: 1.125

**С экспоненциальной частью:**

$1.2 \cdot 10^{23}$

**Рациональные:** можно представить в виде  $m/n$ , где  $m$ - целое число,  $n$  – целое число.

**Иррациональные:** нельзя представить в виде  $m/n$  (число  $\pi$ )

**Комплексные:**  $x+iy$ , где  $i = \sqrt{-1}$

Все эти числа надо каким-то образом представлять в памяти компьютера, выражая их в двоичной системе счисления

# Представление действительных чисел в компьютере

Для представления действительных (вещественных) чисел в современных компьютерах принят способ представления **с плавающей запятой (floating point)**. Этот способ представления опирается на нормализованную (экспоненциальную) запись действительных чисел.

Как и для целых чисел, при представлении действительных чисел в компьютере используется двоичная система, следовательно, предварительно десятичное число должно быть переведено двоичную систему.

**Нормализованная запись** отличного от нуля действительного числа - это запись вида  $a = \pm m \cdot P^q$ , где  $q$  - целое число (положительное, отрицательное или ноль), а  $m$  - правильная дробь. При этом  $m$  называется **мантиссой** числа,  $q$  - **порядком** числа.

# Представление вещественных чисел в компьютере

## Примеры:

- $3,1415926 = 0,31415926 * 10^1$ ;
- $1000 = 0,1 * 10^4$ ;
- $0,123456789 = 0,123456789 * 10^0$ ;
- $0,0000107_8 = 0,1078 * 8^{-4}$ ; (порядок записан в десят. системе)
- $1000,0001_2 = 0,10000001_2 * 2^4$ .

Нормализованная запись нуля в десятичной системе будет такой:  $0 = 0,0 * 10^0$ .

# Представление чисел с плавающей точкой

При представлении чисел с плавающей точкой часть разрядов ячейки памяти отводится для записи порядка числа, остальные разряды - для записи мантиссы. По одному разряду (биту) в каждой группе отводится для изображения **знака порядка** и **знака мантиссы**. Для того чтобы не хранить знак порядка, был придуман так называемый **смещенный порядок**, который рассчитывается по формуле

$$2^{a-1} - 1 + \text{ИП (истинный порядок)}$$

где  $a$  - количество разрядов, отводимых под порядок.

## Пример:

Если истинный порядок равен  $-5$ , тогда смещённый порядок для 4-байтового числа (из которых 1 байт выделен на порядок) будет равен  $2^{8-1} - 1 + (-5) = 128 - 1 + (-5) = 122$ .



# Алгоритм представления числа с плавающей запятой

1. перевести число из  $P$ -ичной системы счисления в двоичную;
2. представить двоичное число в нормализованной экспоненциальной форме;
3. рассчитать смещённый порядок числа;
4. разместить знак, порядок и мантиссу в соответствующие биты ячейки памяти.

## 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).

- Этап 1: Представление числа -25,625 в двоичном виде

Целая часть:

$$25_{10} = 11001_2;$$

Дробная часть:

		0,625
×		2
<hr/>		
		1,250
×		2
<hr/>		
		0,500
×		2
<hr/>		
		1,000

$$0,625_{10} = 0,101_2;$$

$$-25,625_{10} = -11001,101_2.$$

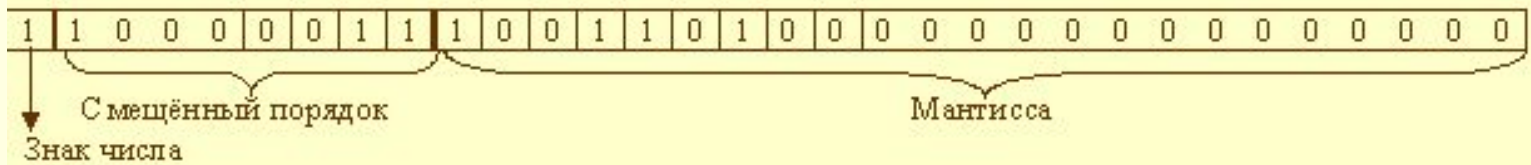
- Этап 2: Преобразование в экспоненциальную форму:

$$-100011,101_2 = -1,00011101_2 \cdot 2^4$$

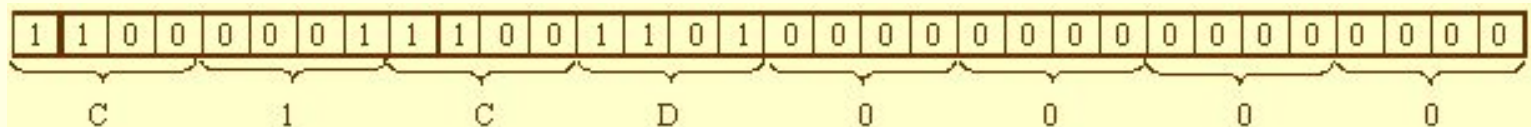
- Этап 3: Расчет смещенного порядка:

$$СП = 127 + 4 = 131 = 10000011_2$$

- Этап 4: Заносим все это в ячейку памяти:



Представление действительного числа не очень удобно изображать в двоичной системе, поэтому часто используют шестнадцатеричное представление:



# Двоичная арифметика - сложение

**Пример.** Сложить двоичные числа  $1101_2$  и  $11011_2$ .

Запишем слагаемые в столбик и пронумеруем разряды, присвоив младшему разряду номер 1:

$$\begin{array}{r} \mathbf{5\ 4\ 3\ 2\ 1} \\ \mathbf{+ 1\ 1\ 0\ 1} \\ \mathbf{1\ 1\ 0\ 1\ 1} \end{array}$$

*разряд 1:*  $1 + 1 = 10$ ; 0 остается в разряде 1, 1 переносится во второй разряд;

*разряд 2:*  $0 + 1 + 1 = 10$ , где вторая 1 - единица переноса; 0 остается в разряде 2, 1 переносится в третий разряд;

*разряд 3:*  $1 + 0 + 1 = 10$ , где вторая 1 - единица переноса; 0 остается в разряде 3, 1 переносится в разряд 4;

*разряд 4:*  $1 + 1 + 1 = 11$ , где третья 1 - единица переноса; 1 остается в разряде 4, 1 переносится в пятый разряд;

*разряд 5:*  $1 + 1 = 10$ ; где вторая 1 - единица переноса; 0 остается в разряде 5, 1 переносится в шестой разряд.

Таким образом:

- 1 1 0 1
- + 1 1 0 1 1
- 1 0 1 0 0 0

Для проверки определим десятичные значения слагаемых и результата:

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = 13;$$

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 8 + 0 + 2 + 1 = 27;$$

$$101000_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 32 + 0 + 8 + 0 + 0 + 0 = 40.$$

Поскольку  $13 + 27 = 40$ , двоичное сложение выполнено верно.

# Двоичная арифметика - вычитание

**Пример.** Вычесть из двоичного числа  $101_2$  двоичное число

$11_2$ .

Запишем алгебраические слагаемые в столбик в порядке "уменьшаемое - вычитаемое" и пронумеруем разряды, присвоив младшему разряду номер 1:

**3 2 1**

■ -1 0 1

■ 1 1

*разряд 1:*  $1 - 1 = 0$ ;

*разряд 2:* поскольку 0 меньше 1 и непосредственное вычитание невозможно, занимаем для уменьшаемого единицу в старшем разряде. Тогда разряд 2 рассчитывается как  $10 - 1 = 1$ ;

*разряд 3:* поскольку единица была занята в предыдущем шаге, в разряде остался 0.

Таким образом:

- 1 0 1
- - 1 1
- 1 0

Для проверки определим десятичные значения слагаемых и результата:

- $101_2 = 5;$
- $11_2 = 3;$
- $10_2 = 2.$

Поскольку  $5 - 3 = 2$ , вычитание выполнено верно.

# Двоичная арифметика - умножение

**Пример.** Умножить двоичное число  $101_2$  на двоичное число  $11_2$ .

Запишем множители в столбик и пронумеруем разряды, присвоив младшему разряду номер 1:

$$\begin{array}{r} \mathbf{3\ 2\ 1} \\ \blacksquare\ 1\ 0\ 1 \\ \blacksquare\ \ \ 1\ 1 \end{array}$$

умножение множимого на разряд 1 множителя дает результат:  
 $101_2 * 1_2 = 101_2$ ;

умножение множимого на разряд 2 множителя дает результат:  
 $101_2 * 10_2 = 1010_2$ . Здесь значение разряда 2 множителя сформировано по принципам формирования значения числа в позиционных системах счисления;

для получения окончательного результата складываем результаты предыдущих шагов:  $101_2 + 1010_2 = 1111_2$ .



# Двоичная арифметика - деление

**Пример.** Разделить двоичное число  $1111_2$  на двоичное число  $11_2$ .

Решение задачи представим схемой:

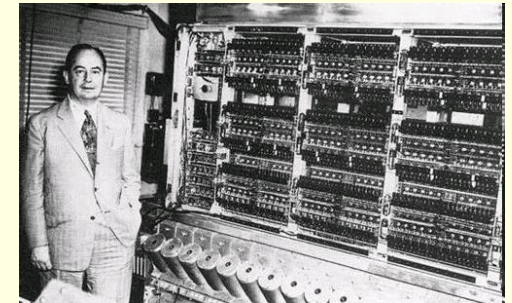
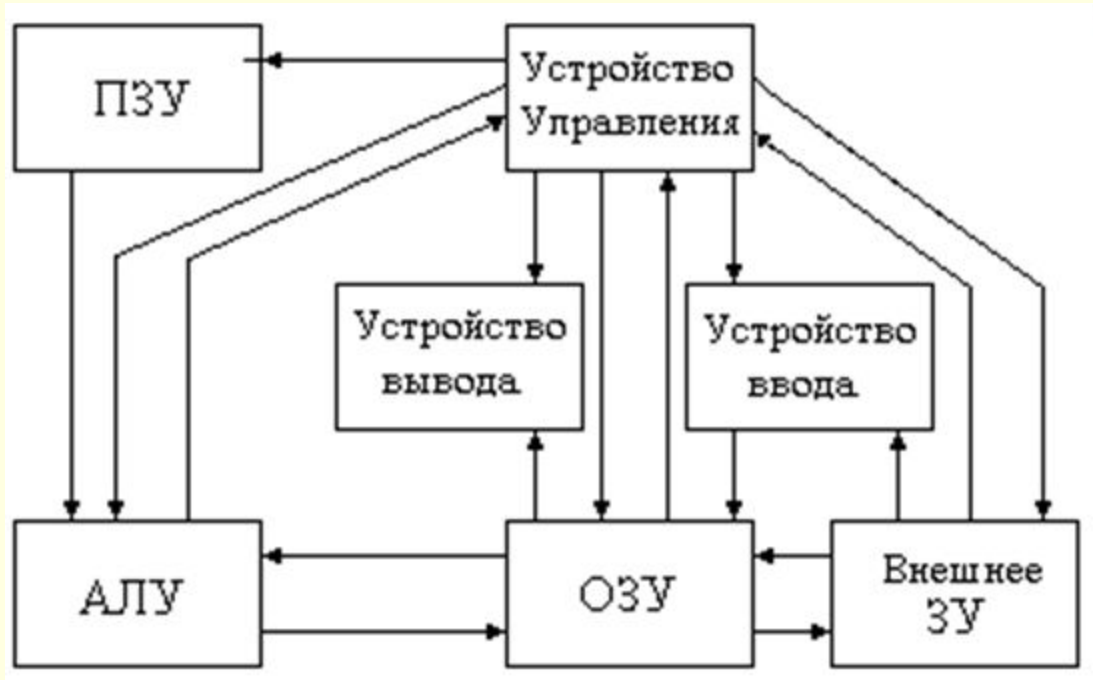
The image shows a handwritten binary division scheme on a yellow background. The dividend is  $1111_2$  and the divisor is  $11_2$ . The quotient is  $101_2$ . The steps are as follows:  $11_2$  goes into  $11_2$  once, leaving a remainder of  $0$ . Then,  $11_2$  goes into  $11_2$  once, leaving a remainder of  $0$ . Finally,  $11_2$  goes into  $11_2$  once, leaving a remainder of  $0$ . The final result is  $101_2$ .

Для проверки правильности результата преобразуем двоичные числа в десятичные:

- $1111_2 = 15;$
- $11_2 = 3;$
- $15 / 3 = 5;$
- $5 = 101_2.$

Деление выполнено верно.

# Как считает компьютер



**Структура компьютера с хранимой программой  
по Дж. фон Нейману**

Все расчеты, которые выполняет компьютер, в конечном итоге базируются всего на одной операции - **сложении**. Вычитание - это сложение уменьшаемого с дополнительным кодом вычитаемого. Умножение - это более сложная операция в виде циклического сложения первого множителя с самим собой, сдвигаемым влево на каждом шаге цикла. Деление - это циклическая комбинация умножений и вычитаний, то есть, в конечном итоге, все тех же сложений.

Все, что должен знать процессор:

$0 + 0 = 0$ ;  $0 + 1 = 1$ ;  $1 + 0 = 1$ ;  
 $1 + 1 = 0$  (перенос)

