Цифровые вычислительные устройства и микропроцессоры приборных комплексов

Сигналы. Аналого-цифровое преобразование.

Способы представления информации

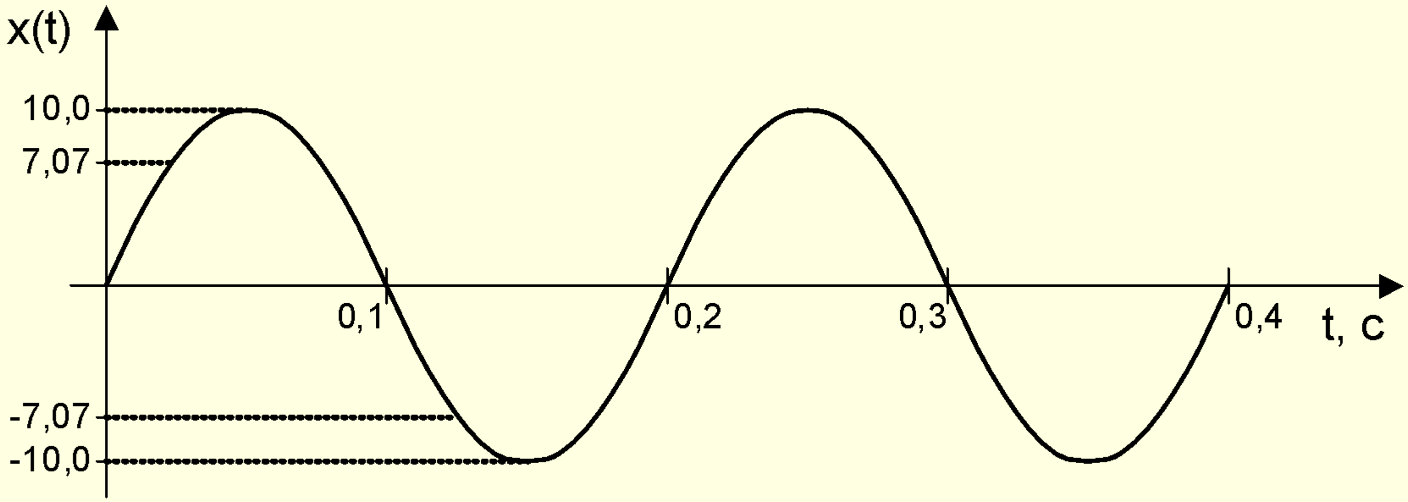
Некрасов Александр Витальевич, канд. техн. наук, доцент кафедры С-16
Соловьёв Сергей Юрьевич, канд. техн. наук, доцент кафедры 303

1. Виды сигналов

- Для представления, передачи и обработки информации в информационных системах используются различные виды сигналов.
- Под *сигналом* понимается физический процесс, значения параметров которого отображают некоторую информацию или сообщение.
- Наиболее распространёнными являются сигналы, представленные в виде электрических колебаний. Информативными параметрами таких сигналов могут быть амплитуда, длительность, частота, фаза и т. д.
- Математически сигнал описывается вещественной или комплексной функцией некоторого вида, определённой на интервале вещественной оси (обычно – оси времени).
- Различают *аналоговые, дискретные и цифровые сигналы*.

Аналоговые сигналы

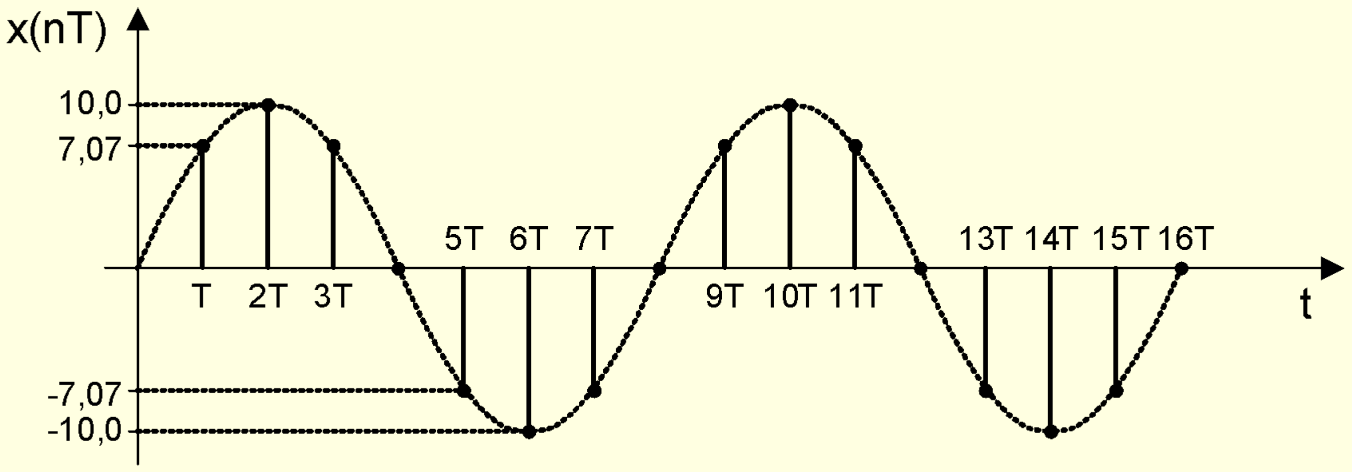
- *Аналоговые сигналы* описываются непрерывной (или кусочно-непрерывной) функцией $x(t)$, причём сама функция и аргумент t могут принимать любые значения на некоторых интервалах $x_1 \le x \le x_2$, $t_1 \le t \le t_2$.
- Пример аналогового сигнала $x(t) = U_m \sin 2\pi ft$ при $U_m = 10$ В, $f = 5$ Гц



Дискретные сигналы

- *Дискретные сигналы* описываются решётчатыми функциями – последовательностями $x(nT)$, где T = const – *интервал дискретизации*, n – целое, $n = 0, 1, 2, \ldots$; функция $x(nT)$ может в дискретные моменты времени nT принимать произвольные значения на некотором интервале. Эти значения функции называются *выборками* или *отсчётами* функции.
- На рисунке показана последовательность отсчётов функции

  $x(nT) = U_m \sin 2\pi f nT$ при $U_m = 10$ В, $f = 5$ Гц, $T = 0{,}025$ с.

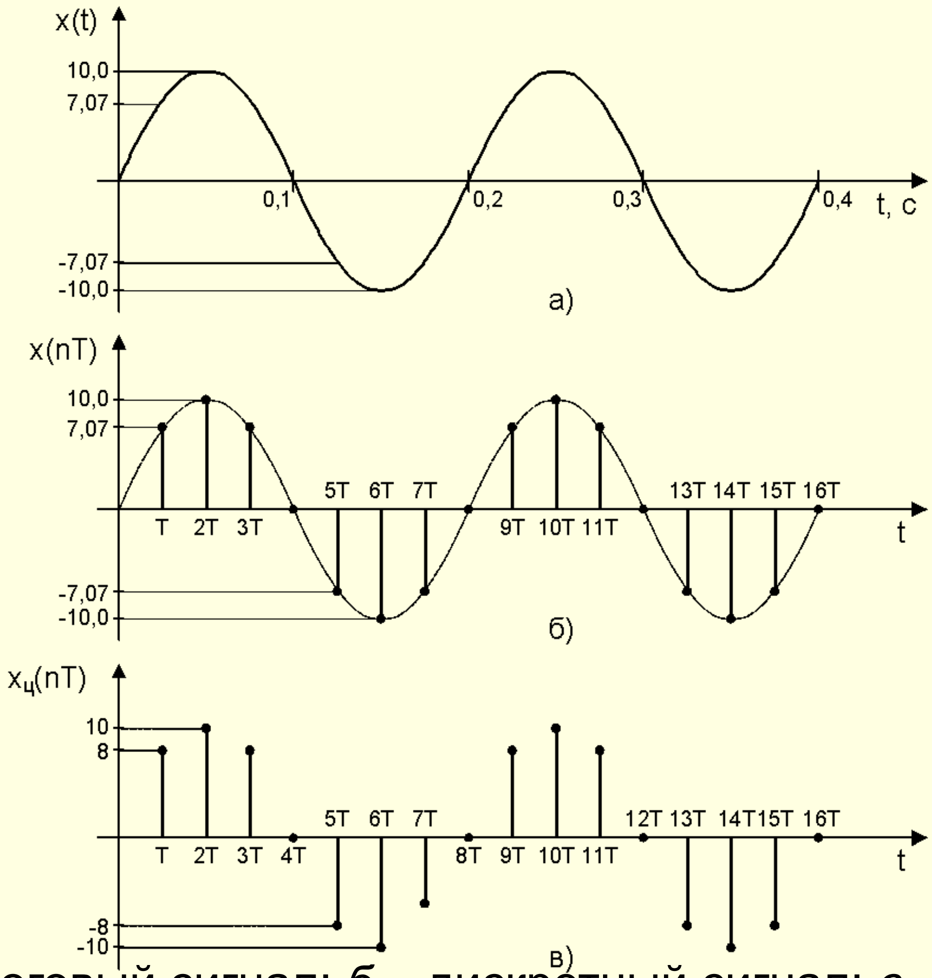


Цифровые сигналы

- *Цифровые сигналы* представляют собой квантованные по уровню дискретные сигналы и описываются квантованными решётчатыми функциями (квантованными последовательностями) $x_ц(nT)$, принимающими в дискретные моменты времени nT лишь конечный ряд дискретных значений – уровней квантования h_1, h_2, …, h_N.
- Связь между решётчатой функцией $x(nT)$ и квантованной решётчатой функцией $x_ц(nT)$ определяется нелинейной *функцией квантования* $x_ц(nT) = F_к(x(nT))$. Существуют различные способы выбора функции квантования. В простейшем случае – при квантовании с постоянным шагом $q = h_l – h_{l–1}$ = const – функция квантования имеет вид

$$x_ц(nT) = F_к\big(x(nT)\big) = \begin{cases} h_1 \ \ при \ \ x(nT) \le (h_2 + h_1)/2; \\ h_l \ \ при \ \ (h_l + h_{l-1})/2 < x(nT) \le (h_{l+1} + h_l)/2; \\ h_N \ \ при \ \ (h_N + h_{N-1})/2 < x(nT). \end{cases}$$

Пример



а – аналоговый сигнал; *б* – дискретный сигнал; *в* – цифровой сигнал

2. Основные операции аналого-цифрового преобразования

Аналого-цифровое преобразование представляет собой совокупность следующих операций:

- **дискретизации** непрерывного сигнала по времени;
- **квантования** дискретных значений сигнала по уровню;
- **кодирования** квантованных дискретных значений сигнала.

- **Дискретизация** по времени связана с временными затратами на выполнение преобразования.
- **Квантование** по уровню вытекает из самого принципа преобразования аналоговой информации в цифровую.
- **Кодирование** представляет собой способ записи результата преобразования.

Дискретизация

- В процессе *дискретизации* из непрерывного сигнала *x*(*t*) берутся отсчёты, которые следуют через определённый временной интервал *T* (интервал дискретизации).
- Переход от непрерывной функции *x*(*t*) к последовательности дискретных значений этой функции математически описывается умножением непрерывной функции на *стробирующую* *r*(*t*), которую можно представить в виде последовательности δ-функций с периодом следования *T*:

$$r(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT).$$

- δ-функция определяется формальным соотношением

$$\int_{-\infty}^{\infty} \delta(x)\,dx = 1 \qquad \delta(x) = 0, \quad \forall x \neq 0$$

- для любой непрерывной функции $f(x)$

$$\int_{\mathbb{R}^n} \delta(x-a) f(x)\,dx = f(a)$$

Интервал дискретизации

- Интервал дискретизации выбирается в соответствии *теоремой Котельникова*, согласно которой, если сигнал имеет ограниченный спектр, т. е. все его спектральные составляющие имеют частоты не выше некоторой частоты f_{max}, то для восстановления аналогового сигнала из последовательности его дискретных значений интервал дискретизации должен удовлетворять условию

$$T \leq \frac{1}{2f_{max}}$$

Квантование

- При *квантовании* непрерывной функции (в рассматриваемом случае значения функции непрерывны в дискретные отрезки времени) непрерывное множество значений функции заменяется эквивалентным множеством дискретных значений, в результате чего образуется ступенчатая функция $y(t)$. Переход с одной ступени на другую теоретически происходит в те моменты, когда функция $x(t)$ пересекает уровень посередине расстояния q между соседними уровнями. Он называется разрешённым уровнем, а само расстояние q представляет собой интервал или *шаг квантования*.
- Квантование по уровню может быть равномерным или неравномерным. Соответственно при квантовании весь возможный диапазон изменения сигнала (от минимального до максимального значения) делится на (n – 1) равных или неравных шагов.

Уравнение идеального квантователя

- Так как при квантовании любое значение функции $x(t)$ округляется до некоторого ближайшего разрешённого уровня $x_i(t)$, то процесс квантования может быть описан равенством

$$x_i = \operatorname{int}\left(\frac{x}{q} \pm \frac{1}{2}\right) q$$

Это выражение называется *уравнением идеального квантователя*.

- В реальных АЦП устройства сравнения и релейные элементы имеют конечные пороги чувствительности, что приводит к появлению некоторой зоны неопределённости их срабатывания и соответствующей погрешности.

Кодирование уровней квантования

- Каждый уровень квантования кодируется числом – цифровым кодом (обычно используются двоичные символы «0» и «1»), соответственно квантованные отсчёты $x_ц(nT)$ записываются в виде m-разрядных двоичных чисел. Число уровней квантования N и наименьшее число разрядов m двоичных чисел, кодирующих эти уровни, связаны соотношением

$$m = \mathbf{int}(\log_2 N),$$

где **int**(A) – наименьшее целое число, не меньшее числа A.

- Например, для представления четырёх уровней квантования ($N = 4$) необходимо $m = 2$ разряда, для $N = 6$ $m = 3$; для $N = 9$ $m = 4$ и т. п.
- Код уровня квантования, соответствующий (с точностью до погрешности квантования) значению преобразуемой величины, представляет собой её цифровой эквивалент.

Методы аналого-цифрового преобразования

- Преобразование аналогового сигнала в соответствующий ему цифровой код может осуществляться различными методами.

- Наиболее распространённые методы следующие:
  - параллельный или мгновенный;
  - последовательного приближения;
  - интегрирования;
  - сигма-дельта.

- Преобразователи, реализующие эти методы, имеют соответствующие названия.

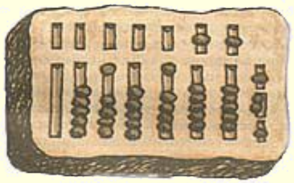
3. Системы счисления

- *Счисление* (нумерация) – совокупность приёмов наименования и обозначения чисел. Под *системой счисления* понимается система правил, позволяющих устанавливать взаимно-однозначное соответствие между любым числом и его представлением в виде совокупности конечного числа символов. Множество символов, используемых для такого представления, называют цифрами.
- В зависимости от способа изображения чисел с помощью цифр системы счисления делятся на **позиционные** и **непозиционные**.
- В **непозиционных** системах любое число есть некоторая функция от численных значений совокупности цифр, представляющих это число. Цифры в непозиционных системах счисления соответствуют некоторым фиксированным числам.

Непозиционные системы счисления

- Примеры непозиционных систем счисления – римская и египетская нумерация.
- Главный недостаток непозиционных систем счисления – неудобство и сложность вычислений, особенно умножения и деления.
- Позиционного нуля нет.
- По мере роста чисел нужны новые обозначения. Невозможно выразить любое число конечным набором знаков.

  - Фактически, ни римская, ни другие непозиционные нумерации не использовалась для реальных сложных вычислений. Для счёта применялись разные виды счетных досок и палочек.
  - На рис.: счётная доска «абак». Расчёты на абаке подобны расчетам в позиционной системе счисления, и потому удобны.



Позиционные системы счисления (1)

- Для представления чисел в цифровых устройствах, а также для представления разнообразной информации в процессе программирования наиболее часто используются **позиционные** системы счисления.
- В позиционных системах счисления числа представляются последовательностью цифр, причём **значение цифры зависит от её положения в числе**. Запись числа разделяется запятой на две группы разрядов, изображающих целую и дробную части числа:

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

- Здесь a_0, a_1, … – цифры 0-го, 1-го и т. д. разрядов целой части числа; a_{-1}, a_{-2}, … – цифры 1-го, 2-го и т. д. разрядов дробной части числа.
- Единице каждого разряда приписан определенный вес p_k, где p – основание системы счисления, k – номер разряда. Таким образом, запись (1) определяет следующее число:

$$N = \ldots + a_2 \cdot p_2 + a_1 \cdot p_1 + a_0 \cdot p_0 + a\_1 \cdot p_{-1} + a_{-2} \cdot p_{-2} + \ldots$$

Позиционные системы счисления (2)

1. Десятичная система счисления (*p* = 10)

- Современная десятичная система счисления возникла в Индии приблизительно в V веке н.э.
- Используется набор из десяти символов: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Например:

$$\mathbf{452{,}361}_{10} = 4 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 6 \cdot 10^{-2} + 1 \cdot 10^{-3}.$$

2. Двоичная система счисления (*p* = 2)

- Описана Лейбницем в XVII веке.
- Используются символы 0 и 1. Например, запись $10111{,}101_2$ соответствует в десятичной системе счисления числу:

$$\mathbf{10111{,}101}_2 = (1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3})_{10} =$$

$$= \mathbf{23{,}625}_{10}.$$

Позиционные системы счисления (3)

3. Восьмеричная система счисления (*p* = 8)

- Широко использовалась в программировании в 1950–1970-х.
- Используются символы 0, 1, 2, 3, 4, 5, 6, 7. Например, запись $537{,}46_8$ соответствует в десятичной системе счисления числу:

$$\mathbf{537{,}46_8} = (5 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0 + 4 \cdot 8^{-1} + 6 \cdot 8^{-2})_{10} = \mathbf{351{,}59375_{10}}.$$

4. Шестнадцатеричная система счисления (*p* = 16)

- Внедрена компанией IBM в 1960-х. Вытеснила восьмеричную.
- Используются символы 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Например, запись $AB9{,}C2F_{16}$ соответствует в десятичной системе счисления числу:

$$\mathbf{AB9{,}C2F_{16}} = (10 \cdot 16^2 + 11 \cdot 16^1 + 9 \cdot 16^0 + 12 \cdot 16^{-1} + 2 \cdot 16^{-2} + 15 \cdot 16^{-3})_{10} = \mathbf{2745{,}761474\ldots_{10}}.$$

4. Способы кодирования информации

- Цифровые устройства используют для представления чисел двоичную систему счисления. При этом наряду с обычным двоичным кодом используются и другие способы кодирования информации:
  - двоично-десятичный код;
  - код Грея;
  - прямой (знаковеличинный код);
  - смещённый код;
  - дополнительный код;
  - корректирующие коды;
  - эффективные коды.

Двоично-десятичный код и код Грея

- В *двоично-десятичном коде* (ДДК, BCD – binary-coded decimal) каждая десятичная цифра записывается в виде группы из 4 двоичных разрядов. Очевидно, что двоично-десятичное кодирование с точки зрения использования двоичных разрядов не экономично, поскольку каждая группа из 4 разрядов способна представлять числа от 0 до 15, а используется для записи числа, не превышающего 9. Тем не менее, ДДК очень удобен для отображения десятичных чисел на индикаторах.
- *Код Грея* (Gray code) формируется таким образом, что при переходе от любого его состояния к следующему изменяется всего лишь один разряд, т. е. код Грея обладает единичным кодовым расстоянием, что используется для исключения ошибок (повышения помехоустойчивости кода). Кодовым расстоянием называется число отличающихся по значению разрядов в паре кодовых слов. Для получения каждого следующего состояния кода Грея выбирается самый младший разряд, изменение которого приводит к образованию нового состояния, и берётся его инверсное значение.

Код Грея и ДДК в сравнении

Десятичная система счисления	*Двоичная система счисления*	*Шестнадцатеричная система счисления*	*Код Грея*	*Двоично-десятичный код (ДДК)*
10 1	**8 4 2 1**	**16 1**		десятки единицы **8 4 2 1 8 4 2 1**
0	0 0 0 0	0 0	0 0 0 0	0 0 0 0 0 0 0 0
1	0 0 0 1	0 1	0 0 0 1	0 0 0 0 0 0 0 1
2	0 0 1 0	0 2	0 0 1 1	0 0 0 0 0 0 1 0
3	0 0 1 1	0 3	0 0 1 0	0 0 0 0 0 0 1 1
4	0 1 0 0	0 4	0 1 1 0	0 0 0 0 0 1 0 0
5	0 1 0 1	0 5	0 1 1 1	0 0 0 0 0 1 0 1
6	0 1 1 0	0 6	0 1 0 1	0 0 0 0 0 1 1 0
7	0 1 1 1	0 7	0 1 0 0	0 0 0 0 0 1 1 1
8	1 0 0 0	0 8	1 1 0 0	0 0 0 0 1 0 0 0
9	1 0 0 1	0 9	1 1 0 1	0 0 0 0 1 0 0 1
1 0	1 0 1 0	0 A	1 1 1 1	0 0 0 1 0 0 0 0
1 1	1 0 1 1	0 B	1 1 1 0	0 0 0 1 0 0 0 1
1 2	1 1 0 0	0 C	1 0 1 0	0 0 0 1 0 0 1 0
1 3	1 1 0 1	0 D	1 0 1 1	0 0 0 1 0 0 1 1
1 4	1 1 1 0	0 E	1 0 0 1	0 0 0 1 0 1 0 0
1 5	1 1 1 1	0 F	1 0 0 0	0 0 0 1 0 1 0 1

Представление чисел со знаком

- Для представления **чисел со знаком** используются прямой (знаковеличинный), смещенный и дополнительный коды.
- В *прямом* (знаковеличинном) *коде* для хранения информации о знаке числа отводится один разряд (обычно старший значащий разряд). Недостатками такого кода являются:
  - операции сложения и вычитания выполняются по-разному (т. е. сложение «не работает» для чисел со знаком)
  - в коде присутствуют нули двух типов («+0» и «–0»)

  Этот код используется при выводе чисел на индикацию, а также в АЦП.
- *Смещённый код*. Для получения смещённого кода некоторого числа необходимо к этому числу, представленному в прямом коде, прибавить половину наибольшего возможного числа. Информацию о знаке здесь также несёт старший значащий разряд, но нуль становится однозначным. Смещённый код используется в АЦП и ЦАП, однако он также неудобен для выполнения вычислений.

Дополнительный код

- В **дополнительном** *коде* положительные числа записываются просто как двоичные без знака, а отрицательные выражаются таким числом, которое, будучи прибавлено к положительному числу той же величины, даст в результате нуль. Дополнительный код также называют дополнением до двух. Чтобы получить отрицательное число, нужно для каждого разряда положительного числа сформировать дополнение до единицы, или обратный код, и затем к полученному результату прибавить 1. Это и даст дополнительный код. Числа в дополнительном коде отличаются от чисел в смещённом коде инверсным значением старшего значащего разряда. Здесь имеется только один нуль, который представляется нулевыми состояниями всех разрядов.
- Арифметические операции над числами в дополнительном коде выполняются согласно обычным правилам двоичной арифметики. Благодаря естественности вычислений дополнительный код наиболее часто применяется для кодирования чисел со знаком в ЭВМ, АЦП, ЦАП и др. цифровых устройствах.

Прямой, смещённый и дополнительный коды в сравнении

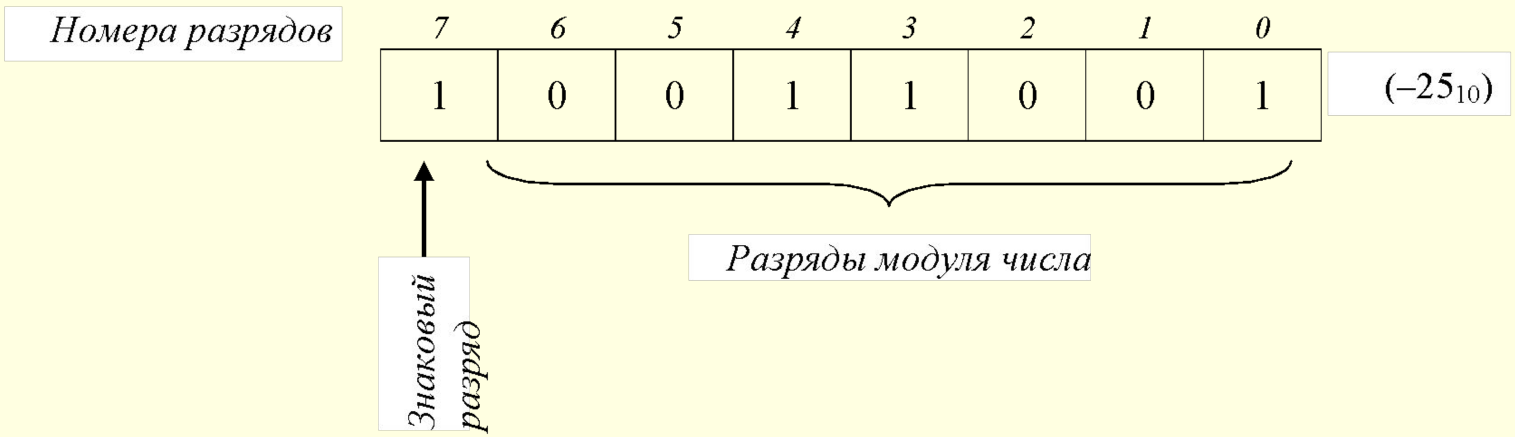
Десятичное число	*Прямой код*	*Смещённый код*	*Дополнительный код*
+7	0 1 1 1	1 1 1 1	0 1 1 1
+6	0 1 1 0	1 1 1 0	0 1 1 0
+5	0 1 0 1	1 1 0 1	0 1 0 1
+4	0 1 0 0	1 1 0 0	0 1 0 0
+3	0 0 1 1	1 0 1 1	0 0 1 1
+2	0 0 1 0	1 0 1 0	0 0 1 0
+1	0 0 0 1	1 0 0 1	0 0 0 1
0	0 0 0 0	1 0 0 0	0 0 0 0
–1	1 0 0 1	0 1 1 1	1 1 1 1
–2	1 0 1 0	0 1 1 0	1 1 1 0
–3	1 0 1 1	0 1 0 1	1 1 0 1
–4	1 1 0 0	0 1 0 0	1 1 0 0
–5	1 1 0 1	0 0 1 1	1 0 1 1
–6	1 1 1 0	0 0 1 0	1 0 1 0
–7	1 1 1 1	0 0 0 1	1 0 0 1
–8	–	0 0 0 0	1 0 0 0
(–0)	1 0 0 0	–	–

5. Форматы представления чисел в цифровых устройствах

- Числа в цифровых устройствах могут представляться в форме целых чисел, чисел с фиксированной запятой и чисел с плавающей запятой.
- **Целые числа** могут храниться в виде знаковых и беззнаковых.
- Целые числа со знаком представляются в знаковеличинном или дополнительном коде.
- Для хранения целых чисел в знаковеличинном коде в ячейке памяти предусматривается следующее распределение разрядов (разрядная сетка): один из *n* разрядов (обычно старший) используется в качестве знакового, в нём в закодированной форме записывается знак числа (0 – в случае положительного числа, 1 – в случае отрицательного числа); остальные разряды используются для хранения модуля числа.

Целые числа в знаковеличинном формате

- Модуль числа занимает в разрядной сетке младшие разряды; свободные старшие разряды заполняются нулями. Например, число -25_{10}, представленное в двоичной системе счисления значением -11001_2, будет размещено в 8-разрядной ячейке памяти следующим образом:



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

Форматы целых чисел в знаковеличинном формате

- В цифровых вычислительных устройствах обычно используется различные форматы целых чисел: с числом разрядов

$n = 8$,

$n = 16$,

$n = 32$,

$n = 64$ и др.

- Максимальные значения модулей чисел для $n = 16$ и $n = 32$ соответственно составляют

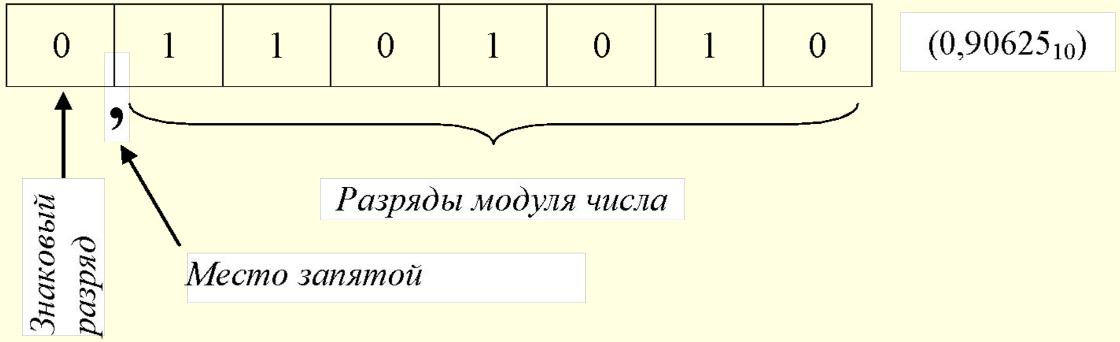
$$2^{15} - 1 \approx 32 \cdot 10^3 \text{ (при } n = 16)$$

- и

$$2^{31} - 1 \approx 2 \cdot 10^9 \text{ (при } n = 32).$$

Числа с фиксированной запятой (ЧФЗ) (1)

- Запятая фиксируется перед старшим разрядом модуля числа.
- Один (старший) разряд используется в качестве знакового; остальные разряды – для хранения абсолютного значения числа.
- **Значение модуля числа всегда меньше единицы.**
- Например, число $0{,}90625_{10}$, представленное в двоичной системе счисления значением $0{,}110101_2$ следующим образом разместится в 8-разрядной ячейке:



Числа с фиксированной запятой (2)

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

$$\varepsilon_{абс} < 2^{-(n-1)}.$$

Так, при $n = 16$ $\varepsilon_{абс} < 2^{-15} = 1 / 32 \cdot 10^3$, при $n = 32$ $\varepsilon_{абс} < 1 / 2 \cdot 10^9$.

Числа с фиксированной запятой (3)

- Так как формат с фиксированной запятой (ФФЗ) предусматривает хранение только дробной части числа, то и исходные данные, и результаты всех проведённых над ними операций должны быть числами, значение которых по модулю меньше единицы. Это обеспечивается выбором определённых масштабных коэффициентов, на которые умножаются исходные данные решаемой задачи. Неправильный выбор коэффициентов может вызвать так называемое **переполнение разрядной сетки**.
- Достоинство чисел в ФФЗ – простота выполнения арифметических операций
- Недостатки чисел в ФФЗ – необходимость масштабирования данных и низкая точность представления данных с малыми значениями модуля (нули в старших разрядах приводят к уменьшению числа разрядов, занимаемых значащей частью модуля числа).

Числа с плавающей запятой (ЧПЗ) (1)

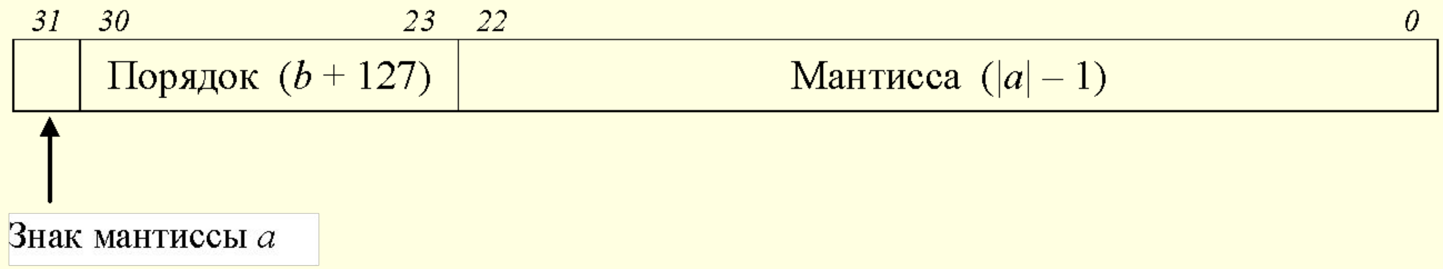
- Формат с плавающей запятой предусматривает представление числа в экспоненциальной форме по основанию 2:

$$a \cdot 2^{b},$$

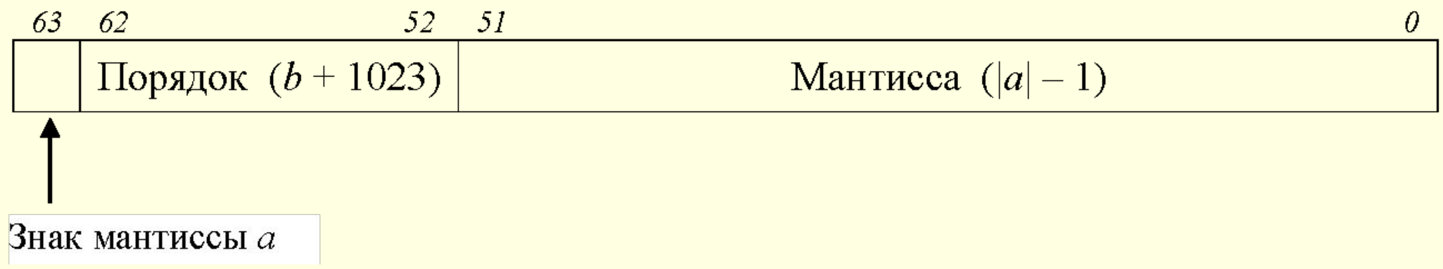
- где *a* – мантисса (от лат. *mantissa* – прибавка), *b* – порядок. Для удобства используется *нормализованное представление*, когда значение модуля мантиссы *a* всегда лежит в диапазоне $1 \le |a| < 2$. В ячейке памяти такие числа хранятся в виде двух групп цифр – мантиссы, которая определяет само число, и порядка, который определяет положение запятой в числе. Таким образом, в данной форме представления чисел запятая является плавающей, что и отражено в названии.
- Форматы представления ЧПЗ определены стандартом **IEEE–754**. Наиболее употребительными являются 2 формы представления ЧПЗ: с одинарной точностью (32 разряда) и с двойной точностью (64 разряда).

Числа с плавающей запятой (2)

Число с плавающей запятой одинарной точности



Число с плавающей запятой двойной точности



Числа с плавающей запятой (3)

Например, число $123{,}456_{10}$ будет следующим образом представлено в формате с плавающей точкой:

– *с одинарной точностью* (32 разряда):

31	30 … 23	22 … 0
0	1 0 0 0 0 1 0 1	1 1 1 0 1 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1 0 0 0

$b + 127 = 10000101_2 = 133_{10}$, отсюда $b = 6_{10}$;
$|a| - 1 = 0{,}11101101110100101111000_2 = 0{,}9289999_{10}$; отсюда $|a| = 1{,}9289999_{10}$;
знак а – «+»; следовательно, записано число $1{,}9289999 \cdot 2^6 = 123{,}4559936_{10}$.

– *с двойной точностью* (64 разряда):

63	62 … 52	51 … 0
0	10000000101	1110110111010010111100011010100111111011111001110111

$b + 1023_{10} = 10000000101_2 = 1102 9_{10}$, отсюда $b = 6_{10}$;
$|a| - 1 = 0{,}1110110111010010111100011010100111111011111001110111_2 = 0{,}9290000000000000_{10}$; отсюда $|a| = 1{,}929_{10}$;
знак а – «+»; следовательно, записано число $1{,}929 \cdot 2^6 = 123{,}456_{10}$.

Числа с плавающей запятой (4)

- Как видно из примеров, ЧПЗ одинарной точности имеют 7 верных десятичных разрядов в дробной части, числа с плавающей запятой двойной точности – 16 десятичных разрядов.
- В формате хранения ЧПЗ:
  - невозможно представить значение 0. Нулевое значение всех битов ЧПЗ одинарной точности представляет величину 2^{-127}. Командами работы с плавающей арифметикой это значение рассматривается как «машинный нуль»;
  - наименьший представимый модуль соответствует $1.17549435 \cdot 10^{-38}$;
  - наибольший представимый модуль $\approx 3.4028235 \cdot 10^{38}$.;
  - «машинной бесконечностью» считается число 2^{128}.