

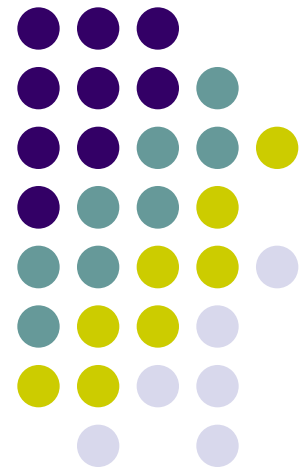
3 Системы счисления. Кодирование чисел целых и дробных

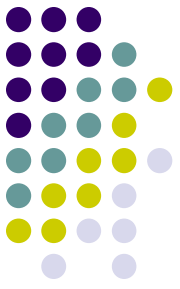
Система счисления - это совокупность правил наименования и изображения чисел с помощью набора знаков.

Системы счисления бывают **позиционные и непозиционные**

Непозиционная система счисления – это система, где порядок цифры в числе определяется по установленному правилу. Например, непозиционной системой счисления является «римская» система.

Позиционной системой счисления, называется система - где порядок цифры в числе определяется рядом степени числа, которое является основанием данной системы счисления.

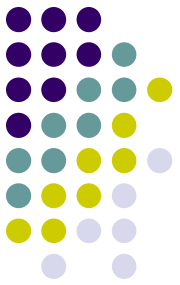




В общем виде целое число в *позиционной* системе счисления можно представить выражением:

$$N(m) = k_0 * m^0 + k_1 * m^1 + \dots + k_{n-1} * m^{n-1}, \text{ где}$$

- $N(m)$ - число в m -ой системе счисления;
- m - разрядность системы ($m = 2$ – двоичная система; $m = 8$ - восьмеричная система; $m = 10$ – десятичная система; $m = 16$ – шестнадцатеричная система);
- n - количество разрядов в числе;
- k - цифра в числе.



Десятичная система счисления

Основанием десятичной системы является ряд степени числа **10**.

Разрядность системы **$m = 10$** .

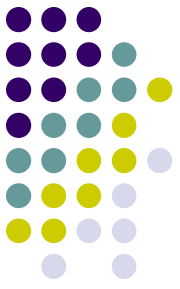
В десятичной системе счисления **10** цифр (от **0** до **9**).

Возьмем, к примеру, десятичное число **1957**.

Число, состоит из четырех цифр - четырехзначное, т.е. **$n = 4$** . Используя выше приведенную формулу, получим число в десятичной системе счисления.

$$N(10) = 7*10^0 + 5*10^1 + 9*10^2 + 1*10^3 = 1957$$

Двоичная система счисления



Основанием **двоичной** системы является **ряд степени числа 2**

Разрядность системы **$m = 2$**

В **двоичной** системе счисления **2** цифры (**0** и **1**)

Возьмем, к примеру, **двоичное** число **100011_B**

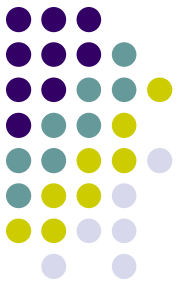
Число, состоит из шести цифр - **шестизначное**, т.е. **$n = 6$**

В – идентификатор **двоичной системы счисления**. Так цифры **0** и **1** используются как десятичной, так и двоичной системой, в записи чисел **недесятичной системы счисления** принято использовать буквы являющиеся **идентификаторами систем счисления** и позволяющие отличить числа одной системы счисления от другой.

Используя выше приведенную формулу, получим десятичное число.

$$N(2) = 1*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 0*2^4 + 1*2^5 = 35,$$

т.е. двоичное число **100011_B** = десятичному числу **35**.



Восьмеричная система счисления

Основанием восьмеричной системы является **ряд степени числа 8**

Разрядность системы **$m = 8$**

В **восьмеричной** системе счисления **8** цифр (от **0** до **7**)

Возьмем, к примеру, **восьмеричное** число **573Q**

Число, состоит из **трех** цифр - трехзначное, т.е. **$n = 3$**

Q – **идентификатор** восьмеричной системы счисления

Используя выше приведенную формулу, получим **десятичное**
число

$$N(8) = 3 * 8^0 + 7 * 8^1 + 5 * 8^2 = 379$$

т.е. восьмеричное число **573Q** = десятичному числу **379**

Шестнадцатеричная система счисления



Основанием шестнадцатеричной системы является ряд степени числа 16

Разрядность системы $m = 16$

В шестнадцатеричной системе счисления 16 цифр (от 0 до F), первые десять цифр от 0 до 9 совпадают с цифрами десятичной системы, а затем идут цифры:

A – цифра десять;

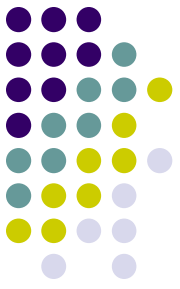
B – цифра одиннадцать;

C – цифра двенадцать;

D – цифра тринадцать;

E – цифра четырнадцать;

F – цифра пятнадцать



Возьмем, к примеру, **шестнадцатеричное** число **1A7H**

Число, состоит из **трех** цифр - трехзначное, т.е. **n = 3**

H – идентификатор **шестнадцатеричной** системы счисления

Используя выше приведенную формулу, получим десятичное число

$$N(16) = 7 * 16^0 + 10 * 16^1 + 1 * 16^2 = 423,$$

т.е. **шестнадцатеричное** число **1A7H** = десятичному числу **423**

Каждый раз, вычисляя число $N(m)$ по приведенной выше формуле мы получаем число в десятичной системе. Таким образом, числа из 2-ой, 8-ой и 16-ой системы мы переводили в десятичную систему счисления.

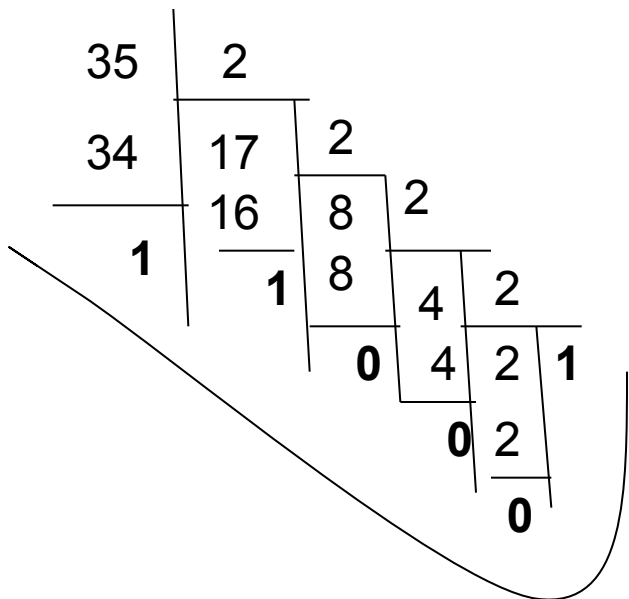
Обратный перевод из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы



Для этого перевода необходимо определить цифры соответствующей системы счисления, которыми будет записано число.

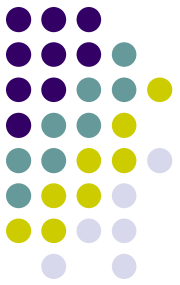
Чтобы получить цифры, которыми будет записано число в системе счисления, необходимо делить столбиком десятичное число на число являющееся основанием системы счисления до тех пор, пока **частное от деления не будет меньше этого числа**, и затем проанализировать результат деления.

Например, как мы выяснили ранее **десятичное число 35** равно **двоичному числу 10011В**. Попробуем его получить. Делим столбиком число 35 на число 2, являющееся основанием двоичной системы счисления.



Анализируя результат деления, складываем полученные цифры системы в число справа налево. Получили двоичное число: **100011_В**. Т.е.

$$35 = 100011_{\text{В}}$$



Десятичное число **379**, как мы ранее узнали равно восьмеричному числу **573Q**. Проверим это. Делим столбиком число 379 на число 8, являющееся основанием восьмеричной системы счисления.

$$\begin{array}{r|l} 379 & 8 \\ \hline 376 & 47 \\ \hline 3 & 40 \\ & \hline & 7 \end{array}$$

5

Анализируя результат деления, складываем полученные цифры системы в число справа налево. Получили восьмеричное число: **573Q**. Т.е.

$$379=573Q$$



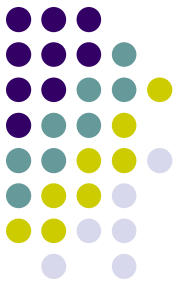
Десятичное число **423**, как мы ранее узнали равно шестнадцатеричному числу **1A7H**. Проверим это. Делим столбиком число 423 на число 16, являющееся основанием шестнадцатеричной системы счисления.

$$\begin{array}{r|l} 423 & 16 \\ \hline 416 & \\ \hline 7 & 26 \\ & \hline & 16 \\ & \hline & 10 \\ & \hline & 1 \end{array}$$

A

Анализируя результат деления, складываем полученные цифры системы в число справа налево. Получили шестнадцатеричное число: **1A7H**. Т.е.

$$423=1A7H$$

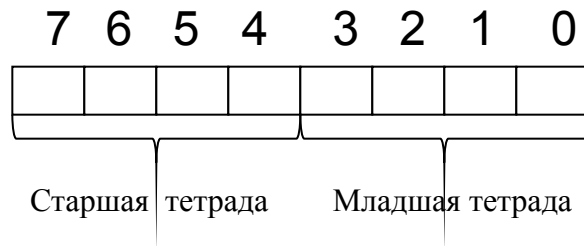


Возникает вопрос, зачем нужны эти системы счисления. Ответ в способах записи информации в компьютер.

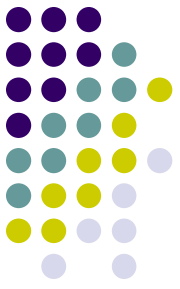
Двоичное кодирование информации, как уже было отмечено ранее, легко реализуется технически. А основу двоичной арифметики обеспечивает группа из *восьми бит* называемая *байт*.

Вся информация в компьютер записывается в байты.

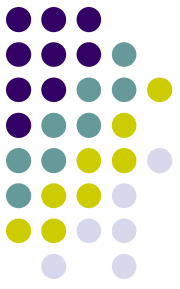
Вид байта с пронумерованными битами



биты в байте нумеруются справа налево



Информация в байты записывается как **числами** целыми и дробными, переведенными в двоичную систему, так и цифрами восьмеричной и шестнадцатеричной системы в так называемых двоично-восьмеричных и двоично-шестнадцатеричных кодах.



Запись информации цифрами

Восьмеричная система

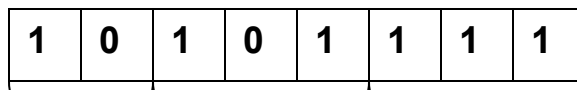
Как известно **цифр** в восьмеричной системе **8** (от **0** до **7**).

Их двоично-восьмеричные коды таковы:

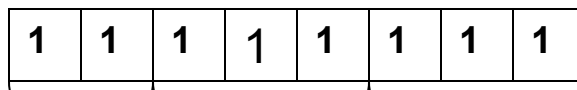
0 – 000В	4 – 100В
1 – 001В	5 – 101В
2 – 010В	6 – 110В
3 – 011В	7 – 111В

Как видим, чтобы записать **одну цифру** восьмеричной системы в двоично-восьмеричном коде необходимо **три бита**.

Поэтому при такой записи информации **байт**, состоящий из восьми битов, **делят не на тетрады по четыре бита**, а на **триады по три бита**



Старшая триада Средняя триада Младшая триада



Видно, что левая **старшая триада** является **усеченной**.

На рисунке в байт записано восьмеричное число **257Q**.

Максимальное восьмеричное число, которое можно записать в **один байт** равно **377Q**.

Запись информации цифрами



Шестнадцатеричная система

Как известно **цифр** в шестнадцатеричной системе **16** (от **0** до **F**).

Их двоично-шестнадцатеричные коды таковы:

0 – 0000В

5 – 0101В

1 – 0001В

6 – 0110В

2 – 0010В

7 – 0111В

3 – 0011В

8 – 1000В

4 – 0100 В

9 – 1001В

A – 1010 – цифра десять

B – 1011 - цифра одиннадцать

C – 1100 - цифра двенадцать

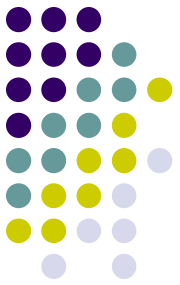
D – 1101 - цифра тринадцать

E – 1110 - цифра четырнадцать

F – 1111 - цифра пятнадцать

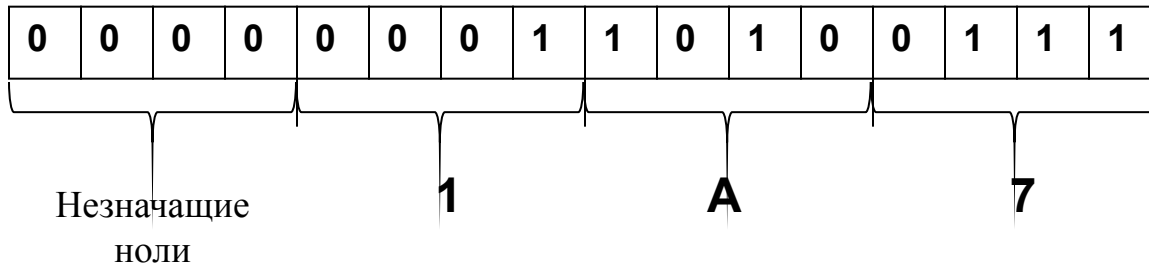
Цифры десятичной системы

Анализируя двоично-шестнадцатеричные коды видно, что, разбивая **байт на тетрады**, и, записывая **одну цифру** двоично-шестнадцатеричного кода **в четырех битах**, мы полностью используем все комбинации двоичных цифр от **0000 до 1111**. Двоично-десятичная же запись информации не позволяет нам это сделать. Поэтому **большинство современных компьютеров используют шестнадцатеричную систему счисления**



Необходимо помнить, что для записи информации **память компьютера выделяется байтами**.

Поэтому, например, полученное нами, в ранее рассмотренных примерах, **число 1A7H** будет записано в память следующим образом.

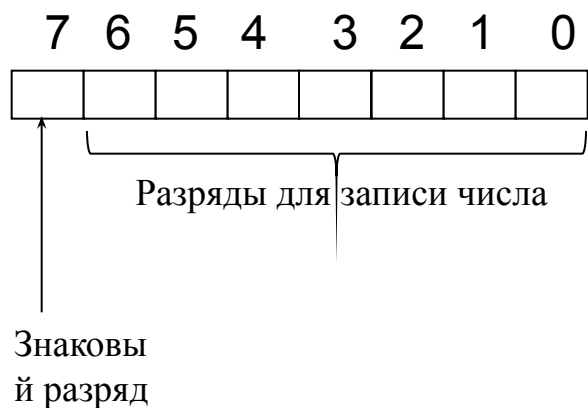


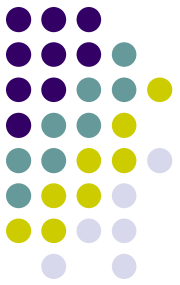
Лишняя тетрада заполняется незначащими нолями



Запись в память компьютера целых десятичных чисел положительных и отрицательных

В этом случае **старший разряд в байте** (или группе байт), выделенном для записи числа, **содержит информацию о знаке числа** и называется **знаковым разрядом**.





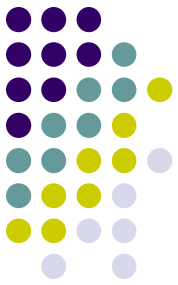
Посмотрим, как записать в память компьютера положительное десятичное число **106**.

Для этого сначала переведем это число в двоичную систему счисления.

Число десятичное **106** равно числу двоичному **1101010_В**.



На рисунке видно, что разряды в байте пронумерованы в соответствии с рядом степени числа 2. В этом порядке и записано в байт наше двоичное число.



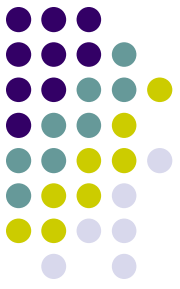
Отрицательные числа в память компьютера записываются в так называемом **дополнительном коде**.

Чтобы получить число в дополнительном коде двоичное число инвертируют и добавляют к нему единицу.

Проинвертировать двоичное число это, значит, записать его разряды наоборот, т.е. вместо единиц записать ноли, а вместо нолей единицы.

0	1	1	0	1	0	1	0	+106
								+
1	0	0	1	0	1	0	1	Инвертированное число
								+
0	0	0	0	0	0	0	1	1
								+
1	0	0	1	0	1	1	0	-106
								+
0	1	1	0	1	0	1	0	+106
								+
1	0	0	1	0	1	1	0	-106
								+
0	0	0	0	0	0	0	0	0

Единица переполнения



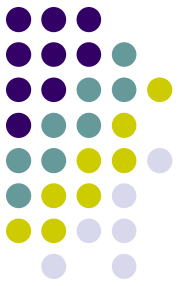
Максимальное целое положительное число которое можно записать в один байт **+127**. **Минимальное отрицательное число**, которое можно записать в один байт **-128**. Для записи чисел, которые не входят в диапазон $-128 \leq x \leq +127$ необходимо использовать два или четыре байта.

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

+127

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

-128



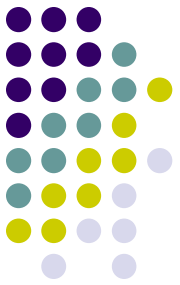
Запись в память компьютера дробных чисел

Познакомимся сначала с понятием *числа с плавающей запятой* или *числа с плавающей точкой*.

Числа с плавающей запятой и числа с плавающей точкой это дробные десятичные числа.

На **Американском континенте** в дробном десятичном числе принято отделять целую часть точкой, а не запятой как у нас в **Европе**. Поэтому в литературе по информатике вам могут встретиться оба эти названия. Так как мы живем в **России**, остановимся на названии числа с плавающей запятой.

Почему запятая плавает?



Возьмем, например, число 2,5.

$$2,5 = 2,5 * 10^0 = 25,0 * 10^{-1} = 0,25 * 10^1$$

Как видим, **запятая плавает** в зависимости от **степени числа 10**. Отсюда название числа с плавающей запятой.

$0,25 * 10^1$ – число в **нормализованном виде**

Число записанное следующим образом: **ноль, запятая, значащая цифра** – называется числом в **нормализованном виде**. Для того, чтобы дробное число было записано в **память компьютера**, оно должно быть приведено в **нормализованный вид**.

Для записи чисел с плавающей запятой в память компьютера используется **полулогарифмическая форма**, которая имеет вид:

$$A = (\pm S) * P^{\pm r}$$

S – мантисса числа **A**, всегда $|S| < 1$.

r – основание системы счисления (целое положительное число);

$\pm r$ – порядок числа

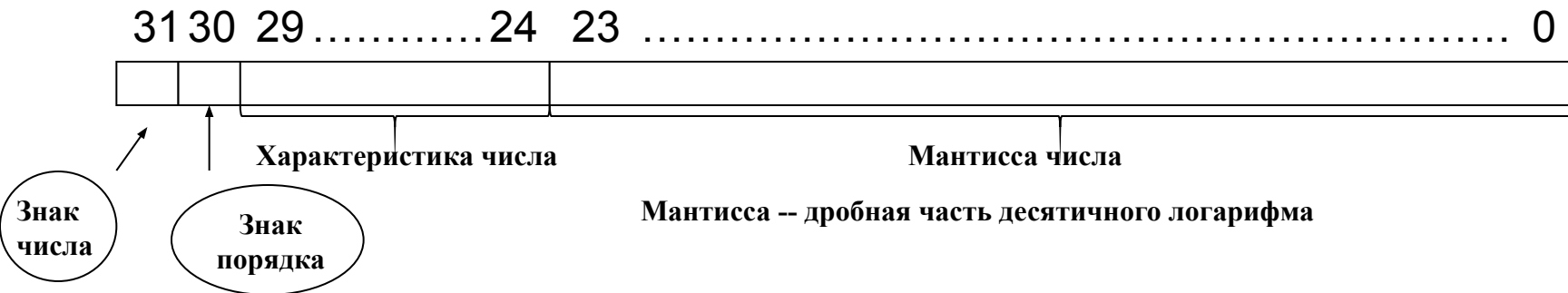
Запись такого типа называется **полулогарифмической**, так как в логарифмической форме представляется часть числа $r^{\pm r}$





Числа с плавающей запятой записываются в память компьютера в четыре или в восемь байтов.

Структура записи числа с плавающей запятой в четыре байта имеет вид:

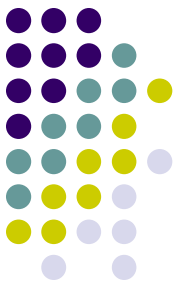


Крайний левый разряд (**31**) содержит знак числа (0 соответствует плюсу, а 1 минусу).

Следующий слева (**30**) разряд знак порядка в числе.

В остальные биты старшего байта (разряды с 29 по 24) записана характеристика числа, она несет в себе информацию о порядке числа.

Остальные биты в оставшихся байтах содержат мантиссу числа в нормализованном виде.



4 Кодирование текстовых, графических и звуковых данных

Текстовые данные

Если каждому символу алфавита сопоставить определенное целое число, то с помощью двоичного кода можно кодировать и текстовую информацию.

Восьми двоичных разрядов достаточно для кодирования 256 различных символов. Этого хватит, чтобы выразить различными комбинациями из восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы, например символ «@».

В настоящее время разработана система, основанная на 16-разрядном кодировании символов. Она получила название UNICODE. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов – этого поля достаточно для размещения в одной таблице символов большинства языков планеты.

Графические данные



Растровая графика

В растровой графике любое графическое изображение состоит из мельчайших точек, образующих характерный узор, называемый *растром*.

Линейные координаты и индивидуальные свойства каждой точки яркость и цветность можно выразить с помощью целых чисел.

Векторная графика

Элементарным объектом векторного изображения является не точка, а линия. Графические объекты создаются как совокупности линий.

В векторных редакторах каждая линия рассматривается как математическая кривая третьего порядка и, соответственно, представляется математической формулой (в компьютере хранятся числовые коэффициенты этой формулы).

Из элементарных объектов (линий) создаются простейшие геометрические объекты (примитивы) из которых, в свою очередь, составляются законченные композиции.

Для оцифровки цветов в современных компьютерах используют две основные модели.



Модель RGB.

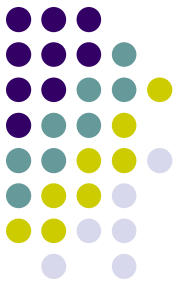
За **основные три цвета приняты красный (Red), зеленый (Green), синий (Blue)**. Каждый составляющий цвет при этом характеризуется своей яркостью, поэтому **модель называется аддитивной**.

Эта схема применяется для создания графических образов в устройствах, излучающих свет, - мониторах, телевизорах.

Модель CMYK.

Изображение формируется на основе отраженной волны от окрашенных поверхностей.

Цвет поверхности можно получить **красителями, которые поглощают, а не излучают**. Например, если мы видим зеленое дерево, то это означает, что из падающего белого цвета, т.е. суммы красного, зеленого, синего, поглощены красный и синий, а зеленый отражен.



Цвета красителей должны быть дополняющими:

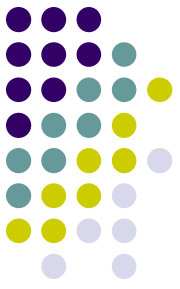
- голубой (Cyan = B+ G), дополняющий красного;
- пурпурный (Magenta = R+B), дополняющий зеленого;
- желтый (Yellow = R+G), дополняющий синего.

Но так как цветные красители по отражающим свойствам не одинаковы, то для повышения контрастности применяется еще черный (black).

Модель **СМΥК** названа по первым буквам слов Cyan, Magenta, Yellow и последней букве слова black.

Так как цвета вычитаются, модель называется **субстративной**.

Кодирование звуков



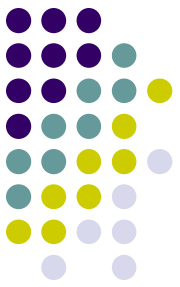
Существует множество методов кодирования звуковой информации.

Звуковые сигналы имеют непрерывный спектр, то есть являются аналоговыми.

Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналого-цифровые преобразователи (АЦП).

Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП).

При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным .



Метод таблично-волнового (Wave-Table) синтеза

В заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментов (и не только для них).

В технике такие образцы называются **сэмплами**.

Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука.

Поскольку в качестве образцов используются «реальные» звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к **качеству звучания реальных музыкальных инструментов.**