Информатика и программирование

Представление информации в ЭВМ

Доцент кафедры ИВТ, к.т.н.
Проскурин Александр Викторович

Содержание лекции

- Двоичное кодирование
- Единицы измерения информации
- Системы счисления
- Представление целых чисел
- Представление чисел с плавающей запятой
- Представление текстовой информации
- Представление графической информации
- Представление аудио- и видео- информации

Кодирование данных

Для автоматизации работы с данными, относящимся к различным типам, очень важно унифицировать их форму представления. Для этого используется **кодирование** – выражение данных одного типа через данные другого типа.

Примером могут служить **естественные человеческие языки** – системы кодирования понятий для выражения мыслей посредством речи.

В истории предпринимались безуспешные попытки создания «универсальных» языков.

Проблема **универсального средства кодирования** достаточно успешно реализуется только в отдельных отраслях техники, науки и культуры.

Например, система записи математических выражений, телеграфная азбука, морская флажковая азбука и другое.

Двоичное кодирование

В вычислительной технике существует своя система – **двоичное кодирование**, основанное на представлении данных последовательностью двух знаков: 0 и 1.

Эти знаки называются двоичными цифрами (binary digit – bit).

- **Бит** – двоичный разряд, принимающий значение 0 или 1.

Одним битом могут быть выражены два понятия: 0 или 1 (да или нет, истина или ложь, черное или белое).

Если количество битов увеличить до двух, то можно выразить четыре различных понятия: 00 01 10 11

Увеличивая на единицу количество разрядов в системе двоичного кодирования, мы увеличиваем в два раза количество значений, которое может быть выражено в данной системе.

Таким образом, если есть **n бит**, можно закодировать **2^n символов**.

Единицы измерения информации

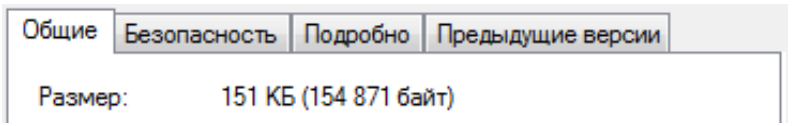
- **Бит** – это минимальная единица измерения количества информации.
- **Байт** является первой более крупной единицей измерения информации: **1 байт = 8 бит = 2^3 бит**.

В компьютере информация кодируется с помощью двоичной знаковой системы, поэтому для измерения количества информации используют коэффициент 2^n. Но так как $10^3 \approx 2^{10}$, то для крупных единиц информации используются те же приставки, что и в системе СИ с коэффициентами 10^n, где n = 3, 6, 9 и т.д.:

1 Килобайт (Кбайт) = 2^{10} байт = 1024 байт.

1 Мегабайт (Мбайт) = 2^{10} Кбайт = 1024 Кбайт = 1 048 576 байт.

1 Гигабайт (Гбайт) = 2^{10} Мбайт = 1024 Мбайт = 1 073 741 824 байт.



Причины использования двоичного кодирования

В компьютере используется двоичное кодирование потому, что оно имеет ряд преимуществ перед другими способами:

- для её реализации нужны технические устройства с двумя устойчивыми состояниями (есть ток – нет тока, намагничен – не намагничен и т.п.), а не, например, с десятью – как в десятичной;
- представление информации посредством только двух состояний надежно и помехоустойчиво;
- возможно применение аппарата булевой алгебры для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

Недостаток двоичной системы – быстрый рост числа разрядов, необходимых для записи чисел.

Системы счисления

Для записи информации о количестве объектов используются **числа**.

Известно множество способов представления чисел. Число изображается символом или группой символов некоторого алфавита. Такие символы называются **цифрами**.

Числа складываются из цифр по особым правилам, которые на разных этапах развития человечества у разных народов были различны. Сегодня их называют системами счисления.

- **Система счисления** – это совокупность приемов и правил для обозначения и именования чисел.

Различают **непозиционные** и **позиционные** системы счисления.

Непозиционные системы счисления

В **непозиционных** системах значение цифры не зависит от положения в числе.

Примером записи чисел в таких системах может служить *римская система*. В качестве цифр в ней используются некоторые буквы латинского алфавита:

Римская цифра	**I**	**V**	**X**	**L**	**C**	**D**	**M**
Значение в десятичной системе	1	5	10	50	100	500	1000

Число представляется как сумма или разность последовательности нужных цифр. Если слева от следующей стоит цифра, соответствующая меньшему количеству, то она вычитается, если справа – прибавляется к числу.

Пример: LXIX = 50 + 10 + 10 - 1 = 69

Позиционные системы счисления

В **позиционных** системах количественное значение цифры зависит от её положения в числе.

Основными характеристиками позиционной системы счисления являются:

- **Алфавит системы счисления** – это совокупность всех цифр, используемых в системе счисления.
- **Основание системы счисления** – количество разных цифр, используемое для представления чисел.
- **Разряд** – позиция цифры в числе.

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

Перевод числа в десятичную систему

Количество, соответствующее числу, можно представить в виде многочлена по степеням основания:

$A_q = a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \ldots + a_0 \cdot q^0$

A – число;
q – основание;
n – число разрядов;
a_i – цифра числа

Например:

$\mathbf{247}_{10} = \mathbf{2} \cdot 10^2 + \mathbf{4} \cdot 10^1 + \mathbf{7} \cdot 10^0$

$\mathbf{1011}_2 = \mathbf{1} \cdot 2^3 + \mathbf{0} \cdot 2^2 + \mathbf{1} \cdot 2^1 + \mathbf{1} \cdot 2^0 = 11_{10}$

Для перевода чисел из любой системы счисления в десятичную необходимо:

1. Представить число в развернутой форме. При этом основание системы счисления должно быть представлено в десятичной системе счисления.
2. Найти сумму ряда. Полученное число является значением числа в десятичной системе счисления.

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

Перевод числа из десятичной системы

Алгоритм перевода целых чисел:

1. Разделить данное число на основание новой системы счисления. Зафиксировать целое частное и остаток от деления.
2. Разделить частное на основание и вновь зафиксировать новое частное и остаток от деления.
3. Повторять шаг 2 до тех пор, пока частное не получится меньше делителя.
4. Записать последнее частное и полученные остатки в обратном порядке в ряд слева направо.

Представление информации в ЭВМ

На сегодняшний день компьютеры способны обрабатывать **числовую, текстовую, графическую, звуковую** и **видео** информацию.

При этом для каждого типа информации был найден способ представления в виде последовательности двоичных цифр. Каждая такая последовательность называется **двоичным кодом**.

Недостаток двоичного кодирования – длинные коды. Но в технике легче иметь дело с большим числом простых однотипных элементов, чем с небольшим числом сложных.

Целые числа без знака

Для представления целых чисел в ЭВМ обычно используют битовые наборы – последовательности нулей и единиц фиксированной длины кратной 8 битам (**формат с фиксированной запятой**).

В таком формате каждому разряду памяти всегда соответствует один и тот же разряд числа.

Для **положительных (беззнаковых) чисел** все биты ячейки памяти участвуют в указании количественного значения числа.

Например, 1 байт равный 8 битам дает возможность задать числа в диапазоне от 0000 0000 до 1111 1111 в двоичной системе (0-255 в десятичной системе).

Целые числа со знаком

Для представления **знаковых целых чисел** используются три способа:

- **Прямой код** – первый (старший) бит показывает знак числа (0 – положительное, 1 – отрицательное), остальные биты отводятся под двоичный код *модуля* числа.
- **Обратный код** – все цифры двоичного кода *модуля* числа, включая разряд знака, инвертируются (0 → 1, 1→ 0).
- **Дополнительный код** – получается прибавлением единицы к обратному коду числа.

Обратный и дополнительный коды позволяют заменить операцию вычитания на сложение.

Целые числа со знаком. Примеры

Битовый набор (k =3)	Беззнаковое целое	Знаковое целое в прямом коде	Знаковое целое в обратном коде	Знаковое целое в дополнительном коде
000	0	+0	+0	+0
001	1	+1	+1	+1
010	2	+2	+2	+2
011	3	+3	+3	+3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

Представление вещественных чисел

Для представления вещественных чисел (содержащих дробную часть) используется **формат с плавающей запятой**.

В этом формате число заносится в память компьютера в **экспоненциальной форме**, то есть в виде двух сомножителей:

- **мантиссы** – дроби, в которой первая значащая цифра стоит сразу после запятой;
- основания системы счисления в соответствующей степени (**порядке**).

Вещественное число	Экспоненциальная форма	Мантисса	Порядок
98567	$0{,}98567 \cdot 10^{5}$	0,98567	5
-98567	$-0{,}98567 \cdot 10^{5}$	-0,98567	5
98,567	$0{,}98567 \cdot 10^{2}$	0,98567	2
-0,0009856	$-0{,}9856 \cdot 10^{-3}$	-0,9856	-3

Представление вещественных чисел (Продолжение)

Таким образом, любое число может быть представлено в виде:

$$N = M \cdot n^p ,$$

где M – мантисса (может быть отрицательной);

n – основание системы счисления;

p – порядок.

В памяти, отведенной для записи числа, выделяются определенные разряды для хранения всех фрагментов числа: знаков мантиссы и порядка, их абсолютных значений.

Пример (код максимального положительного числа):

0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
знак и значение мантиссы														знак и значение порядка							

Представление текстовой информации

Поскольку память компьютера позволяет хранить только числовую информацию, то напрямую работать с текстом невозможно. Чтобы обойти это ограничение были придуманы и активно используются **таблицы кодировки** или просто **кодировки**.

Их суть заключается в том, что все символы алфавита, а также знаки препинания, арифметические и прочие последовательно нумеруются (начиная с 0). То есть **каждому символу присваивается свой код**.

При кодировании текста в память последовательно заносятся полученные коды символов, составляющих текст. То есть если мы определяем числа 47 и 74 как текстовую информацию, коды этих чисел будут отличаться только порядком следования кодов цифр 4 и 7.

Для кодирования текстов могут использоваться различные таблицы перекодировки. Важно, чтобы **при кодировании и декодировании одного и того же текста использовалась одна и та же таблица**.

Таблица кодировок ASCII

Для английского языка институт стандартизации США (ANSI – American National Standard Institute) ввел в действие систему кодирования **ASCII** (American Standard Code for Information Interchange – стандартный код информационного обмена США).

В системе ASCII закреплены две таблицы кодирования: **базовая** и **расширенная**. Базовая таблица закрепляет значения кодов от 0 до 127 (первые 2^7 битов), а расширенная относится к символам с номерами от 128 до 255 (оставшиеся состояния при задействовании 2^8 битов).

Первые 32 кода базовой таблицы, начиная с 0, отданы производителям аппаратных средств. В этой области размещаются так называемые управляющие коды, которыми можно управлять тем, как производится вывод прочих данных.

Начиная с кода 32 по код 127 размещены коды символов английского алфавита (строчные и прописные), знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.

Базовая таблица кодировок ASCII

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Американский код ASCII фактически стал международным стандартом, из-за чего национальным системам кодирования пришлось «отступить» во вторую, расширенную часть системы кодирования, определяющую значения кодов со 128 по 255. Отсутствие единого стандарта в этой области привело к множественности одновременно действующих кодировок.

Кодировки в России

В России наиболее популярны следующие расширенные кодировки:

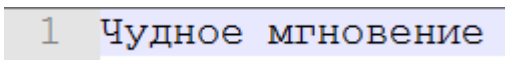
- **CP1251** (Code Page) – кодировка с кириллицей компании Microsoft. Не имеет символов псевдографики. В России она глубоко закрепилась и нашла широкое распространение. Эта кодировка используется на большинстве локальных компьютеров, работающих на платформе Windows.
- **CP866** (альтернативная кодировка) – кодировка с кириллицей компании Microsoft. Используется ею в консоли, поскольку содержит символы псевдографики (позволяют «рисовать» рамки таблиц).
- **КОИ8** (Код Обмена Информацией, Восьмизначный). На базе этой кодировки ныне действуют кодировки КОИ8-Р (русская) и КОИ8-У (украинская). Кодировка КОИ8-Р имела широкое распространение в компьютерных сетях на территории России и в некоторых службах российского сектора Интернета (в сообщениях электронной почты и телеконференций).

Проблемы кодировок

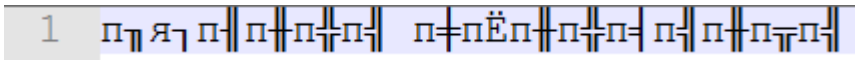
256 состояний 1 байта недостаточно для того, чтобы закодировать все встречающиеся символы и способы форматирования текста, учитывая разнообразие естественных языков, а также фирм, выпускающих программное обеспечение. Это привело к появлению национальных кодовых таблиц.

Однако такое решение ограничивает допустимые символы в пределах одного документа только одной конкретной кодировкой. Это в свою очередь приводит к появлению новых немного измененных таблиц.

Множество используемых кодировок в итоге привело к проблеме возникновения так называемых ***кракозябр*** (*mojibake*). Их можно наблюдать когда текст написан в одной кодировке, а отображать его пытаются в другой.



CP1251



КОИ6-Р

Юникод (Unicode)

Юникод (Unicode) – стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира.

Первая версия Юникода представляла собой кодировку с фиксированным размером символа в 16 бит, то есть общее число кодов было 2^{16} = 65 536. Отсюда происходит практика обозначения символов четырьмя шестнадцатеричными цифрами (например, U+04F0).

При этом в Юникоде планировалось кодировать не все существующие символы, а только те, которые необходимы в повседневном обиходе. Первые 128 состояний в нём полностью совпадают с ASCII.

Применение этого стандарта позволяет закодировать большое число символов из разных систем письменности. При этом становится ненужным переключение кодовых страниц, но длина текста удваивается, а скорость его обработки замедляется.

Плоскости Юникода (кодовое пространство)

В дальнейшем было принято решение кодировать все символы, в связи с чем 65 536 состояний оказалось недостаточно и пришлось значительно расширить кодовую область.

Одновременно с этим коды символов стали рассматриваться не как 16-битные значения, а как абстрактные числа, которые в компьютере могут представляться множеством разных способов. В кодировке UTF-8 предполагается использование от 1 до 4 байтов в зависимости от символа.

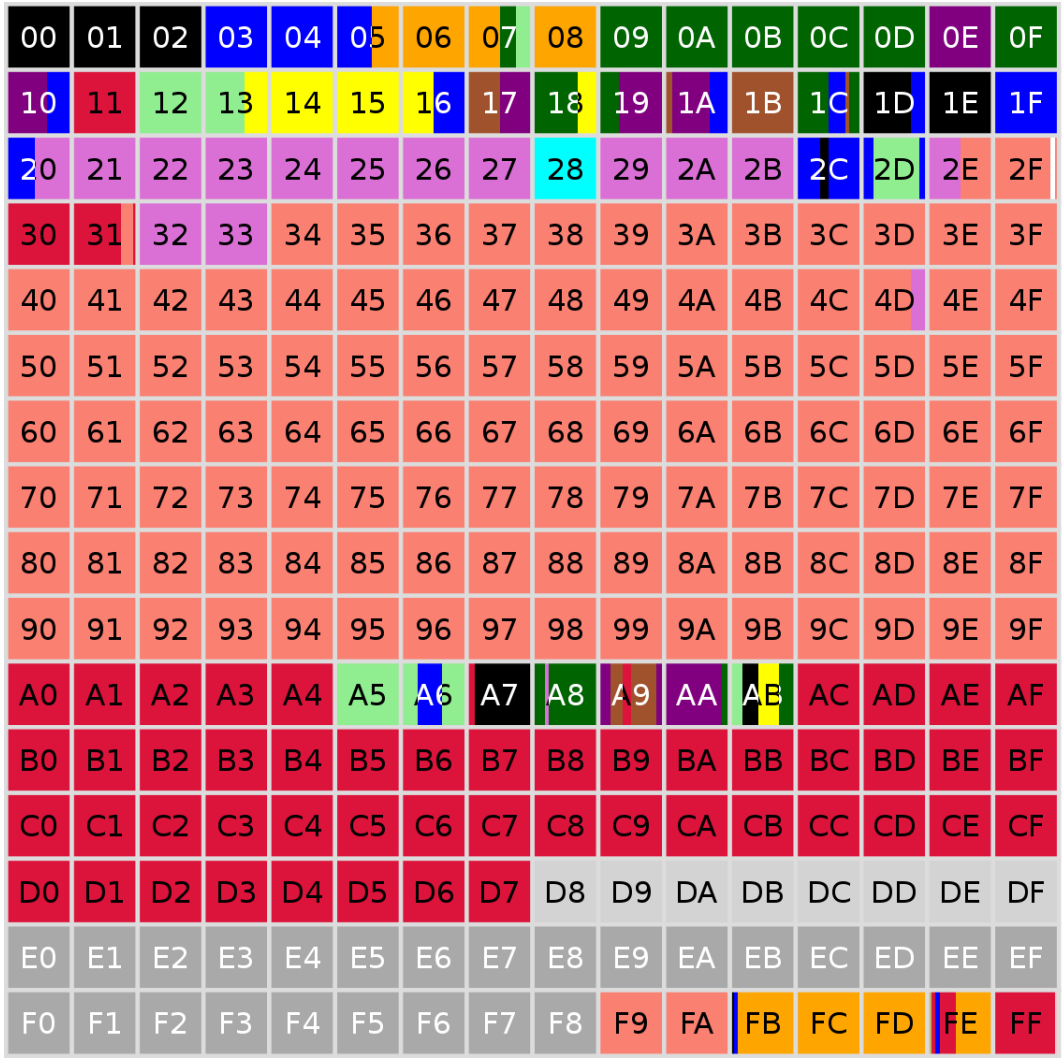
Теперь кодовое пространство разбито на **17 плоскостей** (planes) по 2^{16} (65 536) символов.

Нулевая плоскость (plane 0) называется **базовой** и содержит символы наиболее употребительных письменностей. Остальные плоскости – дополнительные.

Для обозначения символов Unicode используется запись вида «U+*xxxx*» (для кодов 0…FFFF) или «U+*xxxxx*» (для кодов 10000…FFFFF), или «U+*xxxxxx*» (для кодов 100000…10FFFF), где *xx* – шестнадцатеричные цифры.

Например, символ «я» (U+044F) имеет код $044F_{16} = 1103_{10}$.

Основная многоязычная плоскость



- Латинская письменность
- Нелатинские европейские письменности
- Письменности Среднего Востока и Юго-Западной Азии
- Письменности Южной и Центральной Азии
- Письменности Африки
- Письменности Восточной Азии
- Письменности Юго-Восточной Азии
- Письменности Америки
- Письменности Индонезии и Океании
- Знаки
- Системы нотописи
- Идеограммы ККЯ
- Суррогатные пары UTF-16
- Область для частного использования

По состоянию на версию Юникода 14.0

Представление графической информации

Графическая информация в докомпьютерную эпоху регистрировалась и воспроизводилась в **аналоговой форме**. Чертежи, рисунки создавались с помощью сплошных линий и мазков разной величины и цвета.

Дискретное представление графической информации получается за счет разделения всего изображения на строки и колонки. Каждая получившаяся клетка (точка) называется **пикселем** (pixels). Количество строк и колонок – это **разрешение** изображения.

Например, типовые современные разрешения мониторов: 1920×1080, 2560×1440 пикселей. Первым указывается количество колонок, вторым – количество строк в растре.

Каждый **пиксель задаётся двоичным кодом цвета** области изображения. Для **монохромных изображений** цвет одного пикселя кодируется в 1 байте. Это позволяет передать 256 оттенков серого.

Для хранения изображения в память записываются коды пикселей начиная с верхнего левого угла. Затем последовательно направо и вниз строка за строкой. Для хранения изображения 1920×1080 в оттенках серого нужно 2 073 600 байт, плюс несколько байтов для указания разрешения.

Цветовые модели

Цветные изображения могут кодироваться разными способами:

- Система **RGB** (Red, Green, Blue) – удобна для изображений, которые будут отображаться излучающими свет устройствами (например, мониторами). Оттенки цвета создаются смешением лучей трёх базовых цветов разной интенсивности. Под значение интенсивности каждого луча отводится 1 байт, т. е. различают 256 уровней интенсивности. Для совокупности трёх лучей получается $256^3 \approx 16{,}7$ млн. разных вариантов, каждый из которых создает свой оттенок цвета.

- Система **CMYK** (Cyan (голубой), Magenta (пурпурный), Yellow (жёлтый), blacK (чёрный)) – удобна для изображений, которые наблюдаются отраженным светом (например, после печати на бумаге). Он учитывает особенности полиграфии, в которой цвет получается смешением четырёх красок. Для кодирования одного пикселя требуется 4 байта (можно передать $256^4 \approx 4$ млрд. оттенков).

Форматы сжатия изображений

Сжатие изображений – применение алгоритмов сжатия данных к изображениям, хранящимся в цифровом виде.

Сжатие изображений подразделяют на **сжатие с потерями** качества и **сжатие без потерь**.

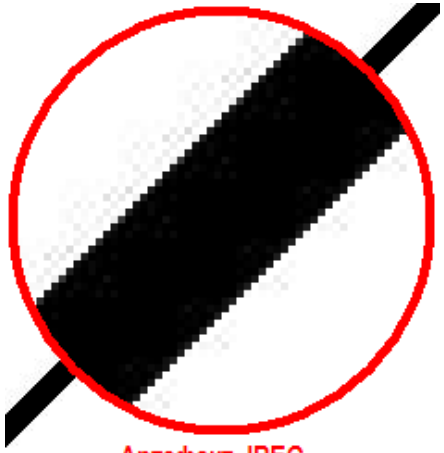
Форматы изображений, в которых применяется сжатие без потерь: BMP, GIF, RAW, PNG и т.д.

Алгоритмы сжатия с потерями при увеличении степени сжатия, как правило, порождают хорошо заметные человеческому глазу артефакты.

Сжатие с потерями применяется в форматах изображений JPEG, JPEG2000 и т.д.

Артефакты JPEG-сжатия

- Потеря чёткости на границах цвета.
- Общая нерезкость.
- Шумовые ореолы вокруг резких границ.
- Блочный эффект.



Артефакт JPEG
серые точки на белом фоне

© Graeme Cookson / Shutha.org

Представление звуковой информации

Звук – это колебания физической среды. В повседневной жизни такой средой является воздух. В электронных устройствах регистрации звука (микрофоне) формируется непрерывно меняющиеся во времени напряжение или ток, т.е. **аналоговый электрический сигнал**.

Для того чтобы преобразовать его в дискретную форму используют специальный блок, входящий в состав звуковой карты компьютера – **аналого-цифровой преобразователь** (АЦП).

Основной принцип его работы заключается в том, что **интенсивность** звукового сигнала **фиксируется** не непрерывно, а **периодически**, в определенные моменты времени.

Представление звуковой информации

Частоту, характеризующую периодичность измерения, называют **частотой дискретизации**. Считается, что для хорошего воспроизведения звука она должна в два раза превышать максимальную частоту волны, входящей в спектр звукового сигнала.

Человеческое ухо воспринимает как звук колебания в диапазоне частот до 22 000 Гц. Следовательно, для хорошего воспроизведения музыки частота дискретизации должна быть не менее 44 000 Гц. Обычная речь воспринимается вполне разборчиво уже при частоте дискретизации 8 000 Гц.

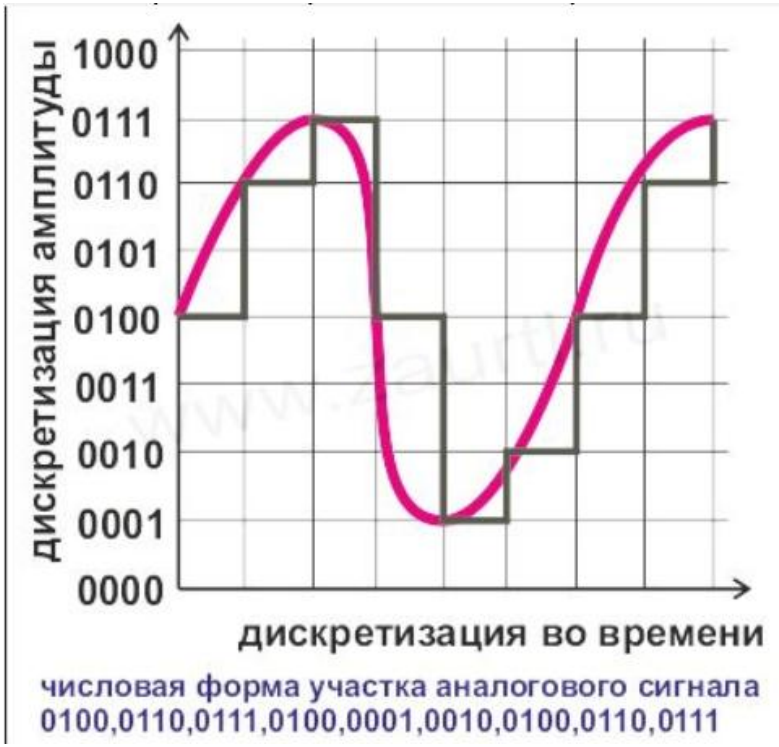
Помимо дискретизации по времени АЦП проводит **дискретизацию и по интенсивности звука**, т.е. по амплитуде звукового сигнала.

В АЦП закладывается сетка стандартных интенсивностей – глубина кодирования (обычно 16 бит или 65 536 уровней) и реальная интенсивность округляется до уровня, ближайшего по сетке.

Представление звуковой информации

Обратное преобразование закодированного таким образом звука в аналоговую форму, воспринимаемую человеческим ухом, производится блоком **цифро-аналогового преобразователя** (ЦАП). По закодированным точкам время-интенсивность с помощью интерполяции рассчитывается гладкая непрерывная кривая, которая используется при восстановлении звукового сигнала.

Для проведения расчетов, восстанавливающих вид звукового сигнала, выпускаются специализированные микропроцессоры, Digital Signal Processor (DSP).



Представление видеоинформации

Видео представляет собой поток последовательно сменяющихся кадров (изображений).

Следовательно, представление видео в ЭВМ сводится к представлению потока графической информации.

Телевизионный формат воспроизведения видео использует разрешение кадра 720*576 точек с 24 битовой глубиной цвета. Скорость воспроизведения составляет 25 кадров в секунду.

Объем передаваемой при этом информации составляет: 24 бита*720*576*25 ≈ 30 Мбайт/с.

Представление видеоинформации

Для хранения и передачи видеоданных используются специальные форматы и алгоритмы сжатия: AVI, MPEG4, H.264, DivX, H.265.

Для того, чтобы компьютер смог распознать и воспроизвести сжатое видео требуется соответствующий видеокодек.

Видеокодек – программа/алгоритм сжатия и восстановления видеоданных.

В отличие от форматов сжатия изображений, сжатие видеоданных использует межкадровую информацию для более эффективной экономии занимаемого места.