

# Представление информации в памяти компьютера

## Компьютерное представление целых чисел

Все числовые данные хранятся в машине в двоичном коде :

- *целые числа хранятся в форме с фиксированной запятой;*
- *вещественные в форме с плавающей запятой ;*
- *текстовая, графическая и звуковая информация кодируется с помощью целых чисел*

1. Для беззнакового представления числа все разряды ячейки отводятся под представление самого числа.
2. Для представления числа со знаком самый старший (левый) разряд отводится под знак числа, остальные разряды – под само число.
3. Если число положительное, то в знаковый разряд помещается 0, если отрицательное – 1.
4. Различие в представлении целых чисел со знаком и без знака вызвано тем, что в ячейках одного и того же размера в беззнаковом типе можно представить больше различных положительных чисел, чем в знаковом

**Например:**

в байте (8 разрядов) можно представить беззнаковые числа от 0 до 255;

1 байт

$$255_{10} = 1111\ 1111_2$$



знаковые – только до 127 :

└─> отводится под знак числа

$$127_{10} = 111\ 1111_2$$



# Прямой код числа

1. Прямой код положительного числа полностью совпадает с записью самого числа

Для хранения целых неотрицательных чисел отводится одна ячейка памяти.

Минимальное число соответствует 8-и нулям, хранящимся в 8-и битах ячейки памяти,

Максимальное значение целого неотрицательного числа достигается в случае, когда во всех ячейках хранятся единицы (число 255):

$$A = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 2^8 - 1 = 255_{10}$$

Для  $n$ -разрядного представления оно будет равно:  $2^n - 1$

**То есть диапазон чисел, которые могут храниться в оперативной памяти в формате целых неотрицательных чисел составляет от 0 до 255 (прямой код)**

Прямой код отрицательного числа отличается от прямого кода соответствующего ему положительного числа содержанием знакового разряда (крайняя левая ячейка)

# Дополнительный код числа

Целые отрицательные числа не представляются в ЭВМ с помощью прямого кода, для их представления используется так называемый дополнительный код, т.е. отводится две ячейки памяти (16 битов), причем старший (левый) разряд отводится под знак числа

Дополнительный код положительного числа равен прямому коду этого числа:

$$2002_{10} = 11111010010_2$$

2 байта=16 битов



1 ячейка

Для  $n$ -разрядного представления оно будет равно:

$$2^{n-1} - 1$$

То есть максимальное число равно  $2^{15} - 1 = 32\,767_{10} = 111111111111111_2$

**Дополнительный код отрицательного числа равен:**

$$A = 2^n - |A|$$

Дополнительный код представляет собой дополнение модуля отрицательного числа  $A$  до 0,

так как в  $n$ -разрядной арифметике.  $2^n - |A| + |A| \equiv 0$  т.е.  $2^n \equiv 0$

/ двоичная запись этого числа состоит из единицы (знаковый разряд) и  $n$  нулей, а в ячейку из  $n$  разрядов может уместиться только  $n$  цифр, в данном случае они все нули /

**Для получения дополнительного кода отрицательного числа можно использовать алгоритм:**

1. Модуль числа записать в прямом коде в  $n$ -двоичных разрядах.
2. Получить обратный код числа, для этого значение всех битов инвертировать (все единицы заменить на нули и все нули заменить на единицы).
3. К полученному обратному коду прибавить единицу.

## Например:

Запишем дополнительный код отрицательного числа «-2002» для 16-и разрядного компьютерного представления:

<b>Прямой код модуля</b>	$ -2002 $	0000 0111 1101 $0010_2$
<b>Обратный код</b>	Инвертирование	1111 1000 0010 $1101_2$
	Прибавление единицы	1111 1000 0010 $1101_2$ 0000 0000 0000 $0001_2$
<b>Дополнительный код</b>	Проверка правильности выполнения алгоритма:	1000 0010 $1110_2$

<b>Прямой код модуля</b>	0000 0111 1101 $0010_2$
<b>Дополнительный код</b>	1111 1000 0010 $1110_2$
<b>Ответ:</b>	1 0000 0000 0000 0000

## Например:

Запишем дополнительный код отрицательного числа «-52» для 8-и разрядного представления:

<b>Прямой код модуля</b>	$ -52 $	$0011\ 0100_2$
<b>Обратный код</b>	Инвертирование	$1100\ 1011_2$
	Прибавление единицы	$1100\ 1011_2$ $0000\ 0001_2$
<b>Дополнительный код</b>		$1100\ 1100_2$

Проверьте правильность выполнения алгоритма.

## Например:

Запишем дополнительный код отрицательного числа «-52» для 16-и разрядного представления:

<b>Прямой код модуля</b>	$ -52 $	0000 0000 0011 0100 <sub>2</sub>
<b>Обратный код</b>	Инвертирование	1111 1111 1100 1011 <sub>2</sub>
	Прибавление единицы	1111 1111 110011011 <sub>2</sub> 0000 0000 0000 0001 <sub>2</sub>
<b>Дополнительный код</b>	Проверьте правильность выполнения алгоритма.	1111 1111 1100 1100 <sub>2</sub>

При  $n$ -разрядном представлении отрицательного числа в дополнительном коде старший

разряд выделяется для хранения знака (единицы). В остальных разрядах записывается

$$2^{n-1} - |A|$$

положительное число:

Чтобы число было положительным должно выполняться условие:

$$|A| \leq 2^{n-1}$$

Следовательно, максимальное значение модуля  $A$  в  $n$ -разрядном

представлении будет равно:  $|A| = 2^{n-1}$

Минимальное отрицательное число равно:

$$A = -2^{n-1}$$

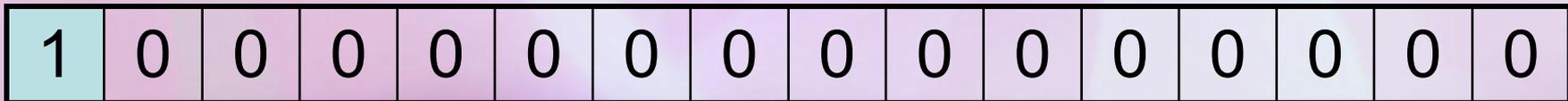
максимальное значение числа со знаком равно  $A=2^{15} - 1 = 32767_{10}$ ,



максимальное значение числа без знака  $A=2^{15} = 65\,535$



минимальное значение числа со знаком равно  $-A = -2^{15} = -32\,768_{10}$



Разрядность	Минимум (без знака)	Максимум (без знака)	Минимум (со знаком)	Максимум (со знаком)
8	0	255	-128	127
16	0	65 535	-32 768	32 767

## Формат длинных целых чисел со знаком

---

Для хранения длинных целых чисел отводится четыре ячейки памяти (32 бита).  
Максимальное число (с учетом выделения одного разряда на знак) равно:

$$A = 2^{31} - 1 = 2147483647_{10}$$

Минимальное отрицательное целое число  
равно:

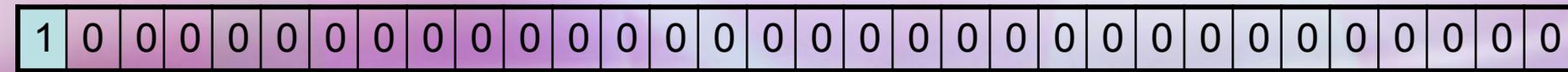
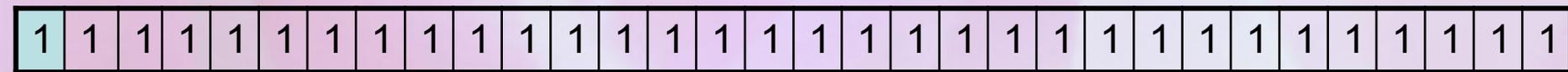
$$A = -2^{31} = -2147483648_{10}$$

Минимальное число без знака –  
0,

максимальное –

$$A = 2^{32} = 4294967295_{10} = 11111111111111111111111111111111_2$$

Разрядность	Минимум (без знака)	Максимум (без знака)	Минимум (со знаком)	Максимум (со знаком)
32	0	4 294 967 295	- 2 147 483 648	2 147 483 647



## ***Достоинства и недостатки представления чисел с фиксированной запятой***

---

**Достоинства:** простота и наглядность представления чисел, простота алгоритмов реализации арифметических действий.

**Недостатки:** небольшой диапазон представления величин, недостаточный для решения математических, физических, экономических и др. задач в которых используются как очень малые, так и очень большие числа.

!!! Выполняя на компьютере вычисления с целыми числами, нужно помнить об ограниченности диапазона допустимых значений. Выход результатов вычислений за границы допустимого диапазона называется переполнением. Переполнение не вызывает прерывания работы процессора. Машина продолжает считать, но результаты могут оказаться неправильными.

!!! Во внутренней памяти компьютера все байты пронумерованы. Нумерация начинается от нуля. Порядковый номер байта называется его адресом. В компьютере адреса обозначаются двоичным кодом. Используется также шестнадцатеричная форма обозначения адреса.

## Представление текстовых данных

Традиционно для кодирования одного символа используется количество информации, равное 1 байту (8 битов). Если рассматривать символы как возможные события, то можно вычислить какое количество различных символов можно закодировать:

$$N = 2^i = 2^8 = 256$$

Кодирование заключается в том, что каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111.

!!! В процессе вывода символа на экран компьютера производится обратный процесс – декодирование, то есть преобразование кода в изображение.

**Вопрос присвоения кода символу –  
соглашение, которое фиксируется в кодовой таблице.**

Первые 33 кода (от 0 до 32) соответствуют операциям: перевод строки, ввод пробела и т.д.

Коды с 33 по 127 являются интернациональными и соответствуют символам латинского алфавита, цифрам, знакам арифметических операций и знакам препинания.

Коды с 128 по 255 являются национальными, то есть в национальных кодировках одному и тому же коду соответствуют различные символы.. В настоящее время существует несколько различных кодовых таблиц для русских букв (КОИ8, ASCII (американский стандартный код для обмена информацией), CP1251, CP866, Mac, ISO...), поэтому тексты, созданные в одной кодировке, не будут правильно отображаться в другой.

**В настоящее время широкое распространение получил новый международный**

**стандарт Unicode, который отводит на каждый символ не один байт, а два,**

**поэтому с его помощью можно закодировать  $2^{16} = 65536$  различных символов. Эту**

**кодировку поддерживают последние версии платформы MS Windows&Office**

**(начиная с 1997 года).**

## Кодировки символов

Двоичный код	Десятичный код	KOI8	CP1251	CP866	Mac	ISO
00000000	0					
...						
0000100	8	Удаление последнего символа (Backspace)				
...						
00001101	13	Перевод строки (Enter)				
...						
00100000	32	Пробел				
00100001	33	!				
...						
10000000	128	-	Ъ	А	А	к
...						
11000010	194	б	В	-	-	Т
...						
11001100	204	л	М	-		Ь
...						
11011101	221	<u>щ</u>	Э	_	Ё	<u>н</u>
...						
11111111	255	<u>ь</u>	я	Нераздел. пробел	Нераздел. пробел	<u>п</u>

# Представление звуковой информации

---

---

Звуковая информация может быть представлена в аналоговой и дискретной форме:

**Аналоговое представление** – физическая величина принимает бесконечное множество значений, причем значения изменяются непрерывно (виниловая пластинка: звуковая дорожка изменяет свою форму непрерывно).

**Дискретное представление** – физическая величина принимает конечное множество значений, изменяется скачкообразно (аудиокомпакт-диск: звуковая дорожка содержит участки с различной отражающей способностью).

Преобразование звуковой информации из аналоговой формы в дискретную производится путем **дискретизации**, т.е. разбиение непрерывного (аналогового) звука на отдельные элементы. В процессе дискретизации производится кодирование, т.е. присвоение каждому звуковому элементу конкретного значения в форме кода

Существует другой способ звукозаписи – **MIDI-запись**: которая является нереальным звуком, а записью отдельных команд-указаний – инструкций синтезатору

# Представление графической информации

В текстовом режиме экран обычно разбивается на 25 строк по 80 символов в строке. В каждую позицию экрана (знакоместо) может быть помещен один символ. Всего на экране  $25 \cdot 80 = 2000$  знакомест.

В каждом знакоместе находится ровно один символ, который может быть высвечен одним из 16-и цветов. При этом можно изменять цвет фона (8 цветов), кроме того, символ может мерцать.

Для представления цвета символа требуется 4 бита ( $2^4=16$ ),

для представления цвета фона - 3 бита ( $2^3=8$ ),

один бит для реализации мерцания (0 – не мерцает, 1- мерцает).

Следовательно, для описания каждого знакоместа требуется 2 байта:

первый байт – символ, второй – его цветовые характеристики.

Таким образом, любой текст или рисунок в текстовом режиме монитора в памяти

компьютера (в видеопамяти) занимает  $2000 \cdot 2 \text{байта} = 4000 \text{байт} = 4 \text{Кбайта}$

**В графическом режиме** экран разделяется на отдельные светящиеся точки (пиксели), количество которых определяет разрешающую способность монитора и зависит от его типа и режима.

Если пиксель не участвует в изображении картинке, то он не светится, если участвует, то светится и имеет определенный цвет. Поэтому каждый пиксель описывается последовательностью нулей и единиц (светится, значит несет цвет, не светится).

Такую форму графических изображений называют растровой.

Используют три основных разрешающие способности экрана:  
800X600, 1024X768, 1280X1024  
и др. (1360X768, 1360X1024, 1440X900, 1440X1200).

## Например:

Рассмотрим формирование на экране монитора растрового изображения, состоящего из 600 строк по 800 точек в каждой строке (всего 480000 точек). В простейшем случае (черно-белое изображение без градаций серого цвета) каждая точка экрана может иметь одно из двух состояний – «черная» или «белая», т.е. для хранения ее состояния необходим 1 бит. Цветное изображение формируется в соответствии с двоичным кодом цвета каждой точки.

Видеопамять	
1	01010101
2	00100101
...	...
800	11110000
...	...
480000	11001100



Цветные изображения могут иметь различную глубину цвета, которая задается количеством битов, которые используются для кодирования цвета точки. Наиболее распространенными значениями глубины цвета являются 8, 16, 24, 32 бита. Каждый цвет можно рассматривать как возможное состояние точки, тогда количество цветов можно вычислить по формуле:

$$N = 2^I$$

где  $I$  – глубина цвета

<b>Глубина цвета (I)</b>	<b>Количество отображаемых цветов (N)</b>
8	$2^8=256$
16	$2^{16}=65\ 536$
24	$2^{24}=16\ 777\ 216$
32	$2^{32}=4\ 294\ 967\ 296$

Цветное изображение на экране монитора формируется за счет смешивания трех базовых цветов: красного, зеленого и синего.

Такая цветовая модель называется **RGB** (Red, Green, Blue).

Для получения богатой палитры цветов базовым цветам могут быть заданы различные интенсивности.

**Например**, при глубине цвета в 24 бита на каждый из цветов выделяется по 8 бит, то есть у каждого из цветов возможны  $N = 2^8 = 256$  уровней интенсивностей, заданные двоичными кодами (от минимальной – 00000000 до максимальной – 11111111).

Название цвета	Интенсивность		
	Красный	Зеленый	Синий
Черный	0000 0000	0000 0000	0000 0000
Красный	1111 1111	0000 0000	0000 0000
Зеленый	0000 0000	1111 1111	0000 0000
Синий	0000 0000	0000 0000	1111 1111
Голубой	0000 0000	1111 1111	1111 1111
Желтый	1111 1111	1111 1111	0000 0000
Белый	1111 1111	1111 1111	1111 1111

## **Например:**

Рассчитать необходимый объем видеопамати для одного из графических режимов с разрешением 800X600 точек и глубиной цвета 24 бита на точку.

## **Решение:**

Всего точек на экране:  $800 \times 600 = 480\ 000$

Необходимый объем видеопамати:

$24 \text{ бит} \cdot 480\ 000 = 11\ 520\ 000 \text{ бит} = 1\ 440\ 000 \text{ байт} = 1406,25 \text{ Кбайт} = 1,37 \text{ Мбайт.}$