

# Архитектура ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Раздел 2  
Информация, кодирование и  
обработка в ЭВМ

# Определение и классификация информации

## Определений информации:

- отрицание энтропии (Л. Бриллюэн);
- мера сложности структур (Моль);
- отраженное разнообразие (Урсул);
- содержание процесса отражения (Тузов);
- вероятность выбора (Яглом);
- снятая неопределенность наших знаний о чем-то (К. Шеннон);
- обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств (Н. Винер).



*Информация* — это осознанные  
сведения об окружающем мире.

# Аналоговая информация

**Исторически первой** технологической формой получения, передачи, хранения информации являлось

**аналоговое** (непрерывное) представление звукового, оптического, электрического или другого сигнала (сообщения).

Магнитная аудио- и видеозапись, фотографирование, запись на виниловые грампластинки, проводное и радиовещание — основные способы хранения и передачи информации в аналоговой форме.

# Аналоговая информация

**Основные понятия: линейность, нелинейность, шум, спектр сигнала, полоса пропускания и пр.).**

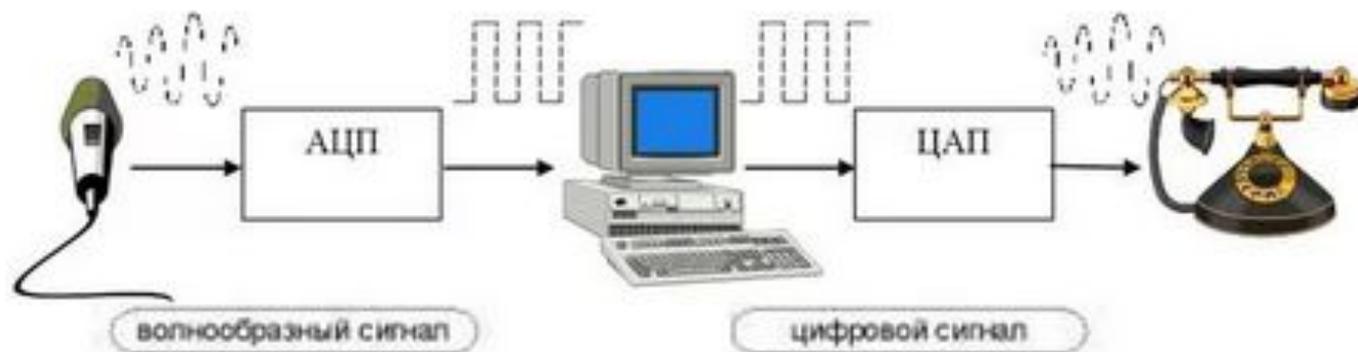


# Аналоговая информация

**Аналого-цифровые (АЦП или дискретные) преобразователи чаще всего изготавливаются в виде интегральных схем.**

**В необходимых случаях осуществляется обратное — дискретно-аналоговое или цифроаналоговое преобразование — ЦАП (digital-to-analog conversion — DAC).**

**Аналоговая** информация, в отличие от цифровой (**дискретной**), непрерывна, волнообразна. Для ввода аналоговой информации в ПК применяются - аналогово-цифровые преобразователи (**АЦП**). Для обратного вывода используют - цифро-аналоговые преобразователи (**ЦАП**). Процесс кодирования звука и изображения называется дискретизацией (или оцифровкой).



# Цифровая информация

Дискретные сигналы по сравнению с

аналоговыми имеют ряд важных **преимуществ**:

- **помехоустойчивость,**
- **легкость восстановления формы,**
- **простоту аппаратуры передачи.**



# Аналоговая и дискретная форма представления информации

## Графическая информация

### Аналоговая форма



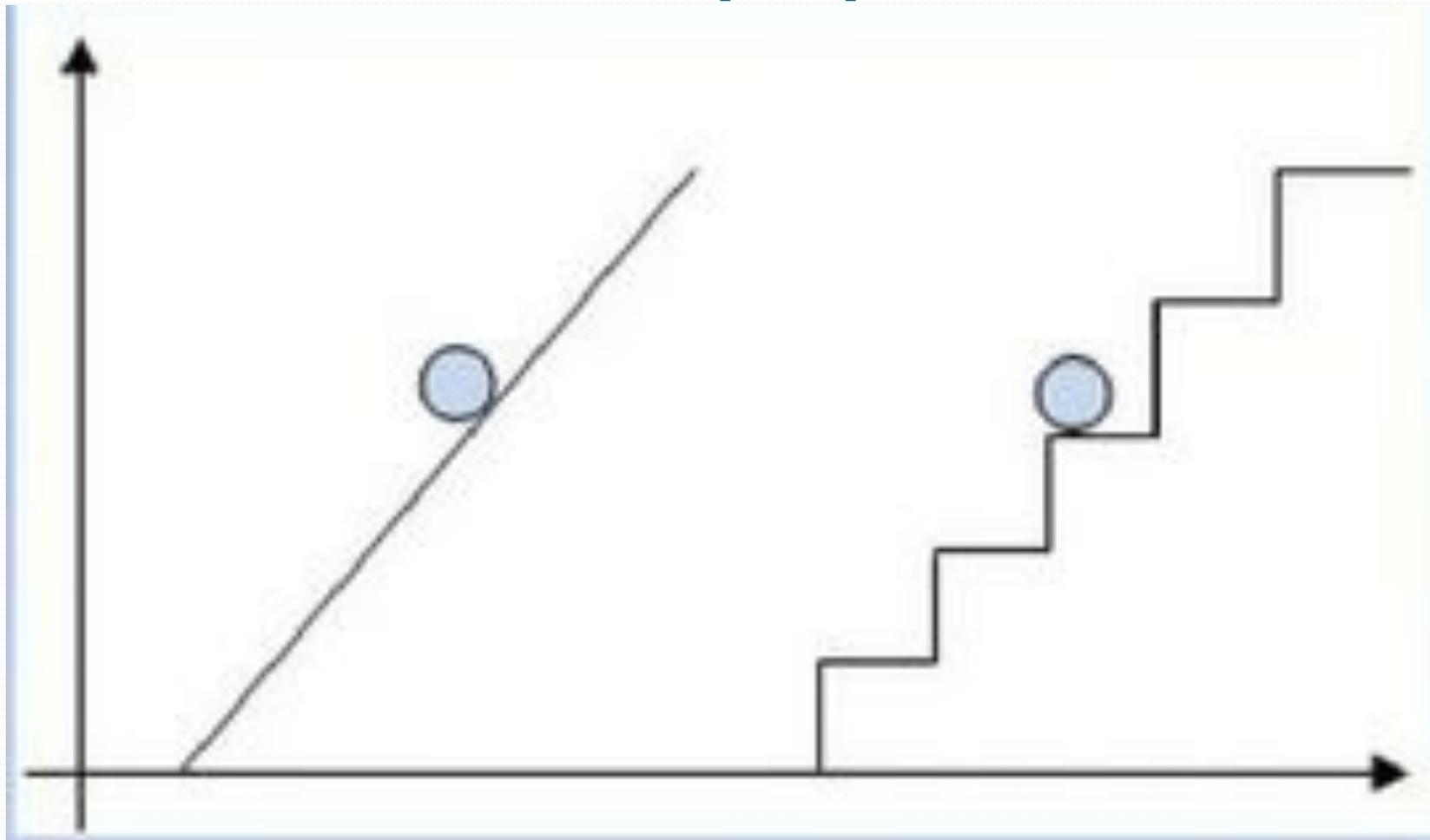
Живописное полотно, цвет которого  
изменяется непрерывно

### Дискретная форма



Изображение, напечатанное с помощью  
струйного принтера и состоящее из 2  
отдельных точек разного цвета

# Какой вид информации?



# Цифровая информация

Сигналы представляют собой дискретные электрические или световые импульсы. При таком способе вся емкость коммуникационного канала используется для передачи одного сигнала. Цифровой сигнал использует всю полосу пропускания кабеля.

**Полоса пропускания** - это разница между максимальной и минимальной частотой, которая может быть передана по кабелю. Каждое устройство в таких сетях посылает данные в обоих направлениях, а некоторые могут одновременно принимать и передавать.

Дискретный цифровой сигнал сложнее передавать на большие расстояния, чем аналоговый сигнал, поэтому его предварительно **модулируют** на стороне передатчика, и **демодулируют** на стороне приёмника информации.

Использование в цифровых системах **алгоритмов проверки и восстановления** цифровой информации позволяет **существенно увеличить** надёжность передачи информации.

**Модуля́ция** (лат. *modulatio* — размеренность, ритмичность) — процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала.

# ПЕРЕДАЧА ДАННЫХ

**Передача данных** — процесс переноса информации в виде сигналов от точки к точке или от точки к нескольким точкам средствами электросвязи по каналу передачи данных.

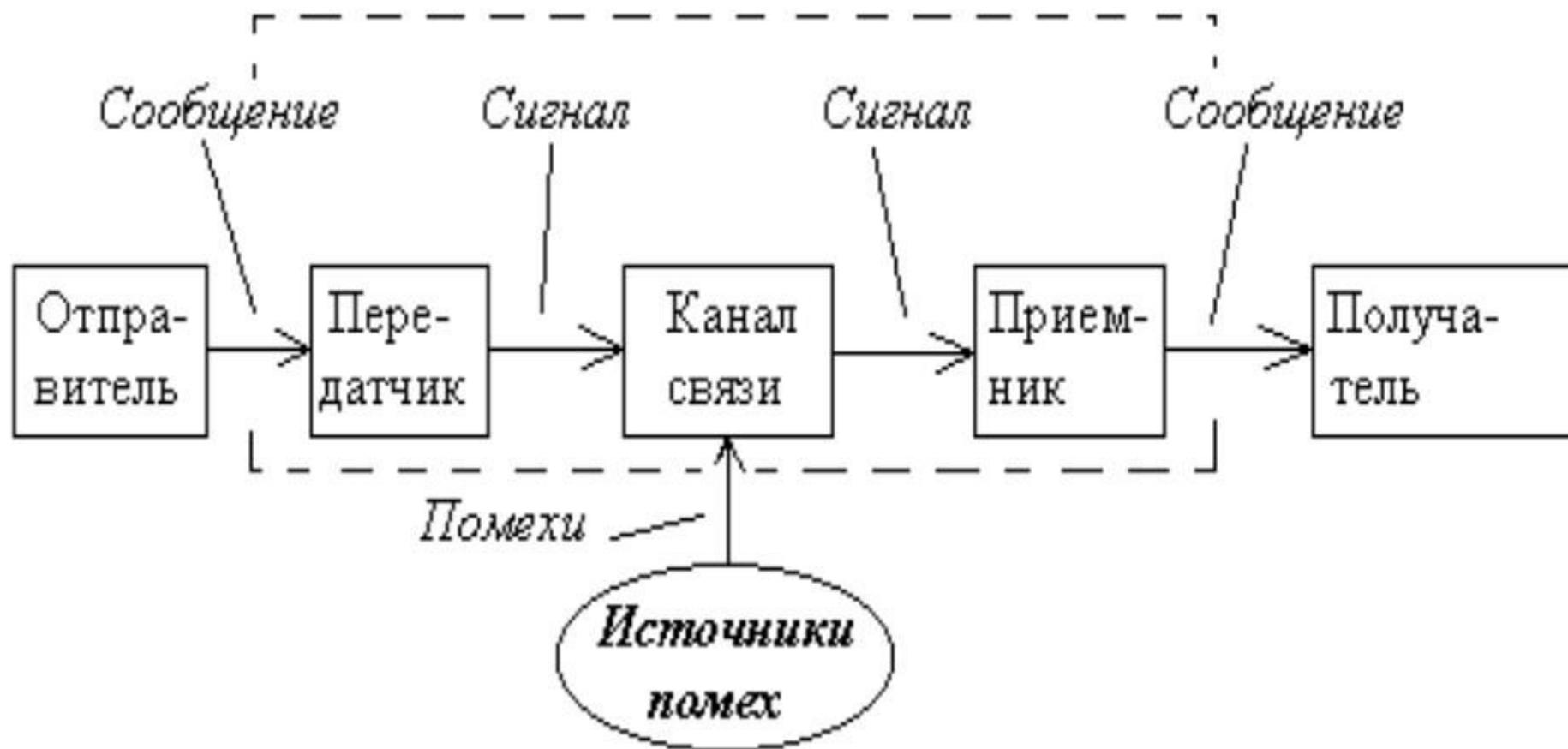


**Канал передачи** — это комплекс технических средств и среды распространения, обеспечивающий передачу сигнала.

При обмене данными по каналам используется три метода передачи данных:

- **симплексная (однонаправленная)** передача (телевидение, радио);
- **полудуплексная** (прием и передача информации осуществляются поочередно);
- **дуплексная (двунаправленная)**, каждая станция одновременно передает и принимает данные.

# Система связи



Широко используются следующие методы последовательной передачи — **асинхронная и синхронная** .

При асинхронной передаче каждый символ передается отдельной посылкой. При синхронной передаче символы группируются.

Стартовые биты **предупреждают приемник** о начале передачи. Затем передается символ.

Для определения достоверности передачи используется бит четности (бит четности равен 1, если количество единиц в символе нечетно, и 0 — в противном случае).

Последний бит («стоп-бит») сигнализирует об окончании передачи.

# Передача данных



*a*



*b*

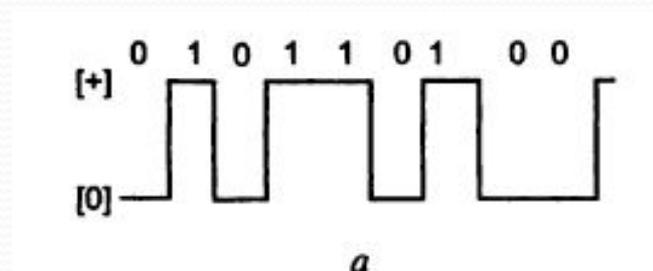
Асинхронная (а) и синхронная (б) передача данных

# Передача данных

При использовании **синхронного метода** данные передаются **блоками**.

Для согласования работы приемника и передатчика в начале блока передаются **биты синхронизации**. Затем передаются данные, код обнаружения ошибки и **символ окончания передачи**.

При **синхронной передаче** данные могут передаваться и как символы, и как поток битов. В качестве кода обнаружения ошибки обычно используется **код CRC**.



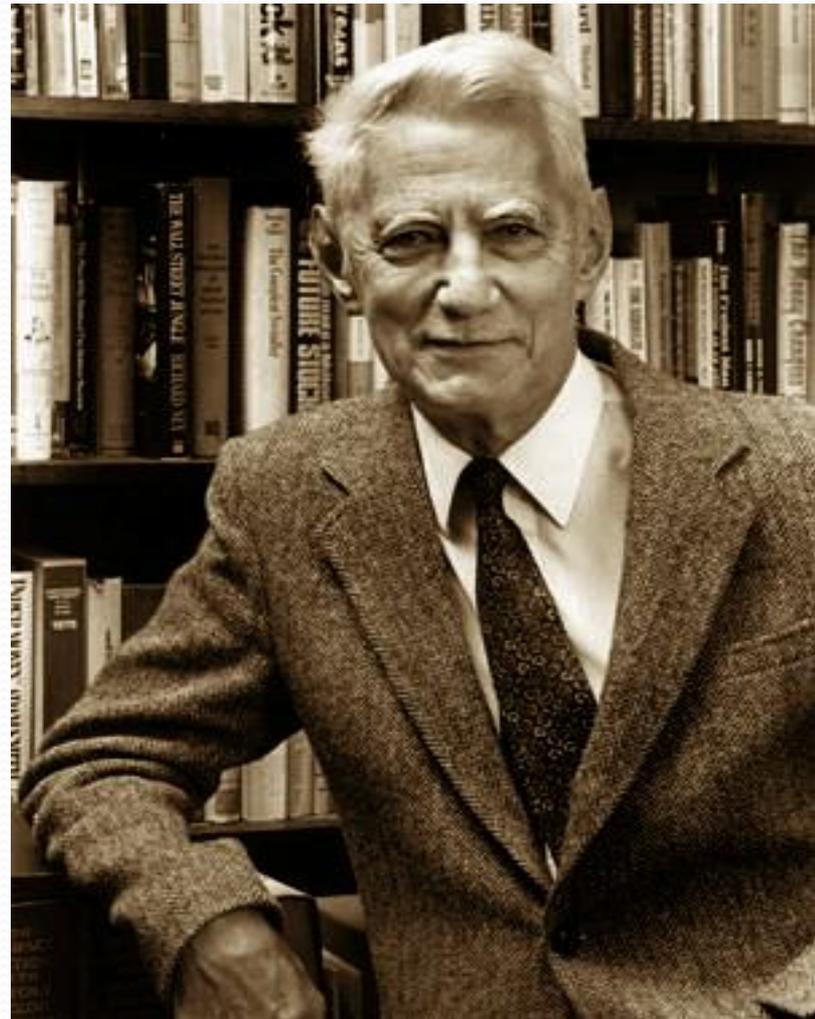
# Контрольные вопросы

1. Что такое информация?
2. Что такое передача информации?
3. По чему происходит передача информации?
4. Какие методы передачи информации существуют?
5. Что применяется для обеспечения корректности передачи информации?
6. Для чего применяются стартовые биты?

# Измерение количества информации

Информацию, содержащуюся в сообщении, можно трактовать в смысле ее новизны или, иначе, уменьшения неопределенности знаний «приемника информации» об объекте.

В качестве единицы информации **Клод Шеннон** предложил принять один **бит** (от англ. bit — binary digit — двоичная цифра).

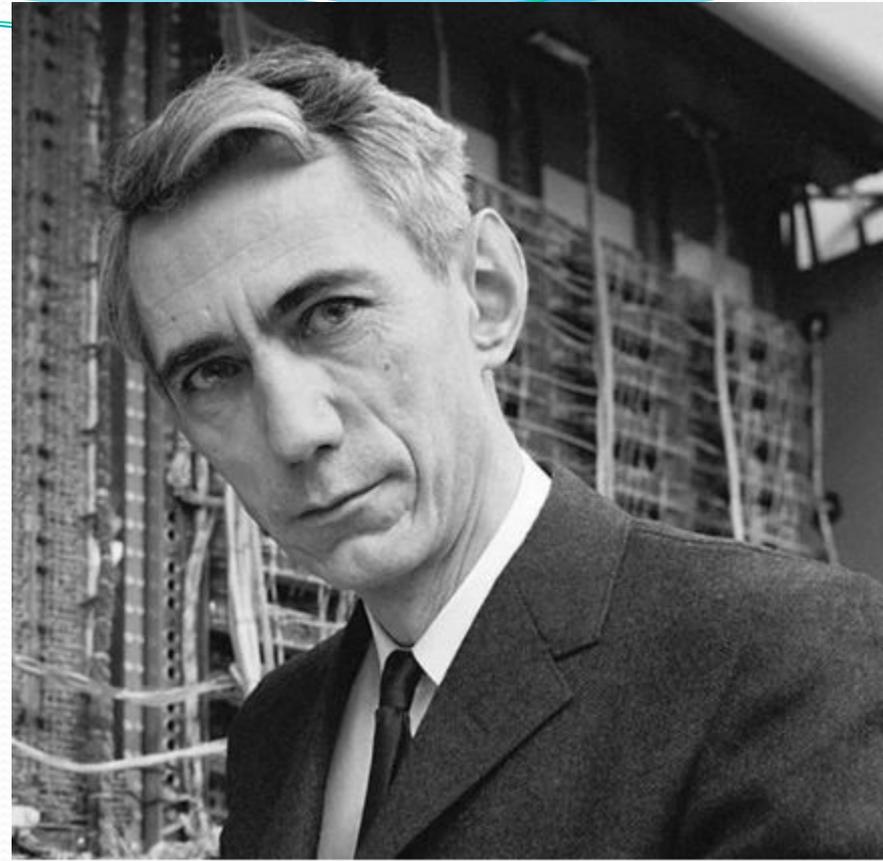


(1916-2001)

# Клод Шенон

Американский инженер, криптоаналитик и математик.

Считается «отцом информационного века». Является основателем теории информации, нашедшей применение в современных высокотехнологических системах связи.



**Предоставил фундаментальные понятия, идеи и их математические формулировки, которые в настоящее время формируют основу для современных коммуникационных технологий.**

**В 1948 году предложил использовать слово «бит» для обозначения наименьшей единицы информации.**

Кроме того, понятие энтропии было важной особенностью теории Шеннона

Магистерскую диссертацию Шеннона по электротехнике Массачусетского технологического института «Символический анализ релейных и переключательных схем» назвали самой важной в 20-м веке.

**Бит** — слишком мелкая единица измерения, на практике чаще применяется более крупная единица — **байт**, равная восьми битам.

В частности, восемь бит требуется для того, чтобы закодировать любой из 256 символов **ASCII** ( $256 = 2^8$ ).

Часто возникает неоднозначность в интерпретации, например, 1 Мбайта, который может рассматриваться как 1000 Кбайт (десятичный мегабайт), так и 1024 Кбайт (бинарный мегабайт).

# Кодирование символьной информации

***Код (code)*** — набор знаков, символов и правил представления информации.

Рассмотрим методы дискретного представления информации, или кодирования (которые, надо сказать, появились задолго до вычислительных машин).

Широко известным примером является азбука Морзе в которой буквы латиницы (или кириллицы) и цифры кодируются сочетаниями из «точек» и «тире».

# ASCII и UTF-8

**ASCII** — American Standard Code for Information Interchange.

**ASCII** была разработана (1963 год) для кодирования символов. ASCII использует для кодирования 8-ми бит. Значит ей можно закодировать **256** символов.

Кодировка символов часто называется **кодовой страницей**.

На смену ей пришла кодировка **UTF-8** (*Unicode Transformation Format*).

В Unicode используются 16-битовые (2-байтовые) коды, что позволяет представить **65536** символов.

# Таблица кодировки ASCII

символ	10- й код	2-Б код	символ	10- й код	2-Б код	символ	10-Б код	2-Б код	символ	10-Б код	2-Б код
	32	00100000	8	56	00111000	P	80	01010000	h	104	01101000
!	33	00100001	9	57	00111001	Q	81	01010001	i	105	01101001
"	34	00100010	:	58	00111010	R	82	01010010	j	106	01101010
#	35	00100011	;	59	00111011	S	83	01010011	k	107	01101011
\$	36	00100100	<	60	00111100	T	84	01010100	l	108	01101100
%	37	00100101	=	61	00111101	U	85	01010101	m	109	01101101
&	38	00100110	>	62	00111110	V	86	01010110	n	110	01101110
'	39	00100111	?	63	00111111	W	87	01010111	o	111	01101111
(	40	00101000	@	64	01000000	X	88	01011000	p	112	01110000
)	41	00101001	A	65	01000001	Y	89	01011001	q	113	01110001
*	42	00101010	B	66	01000010	Z	90	01011010	r	114	01110010
+	43	00101011	C	67	01000011	[	91	01011011	s	115	01110011
,	44	00101100	D	68	01000100	\	92	01011100	t	116	01110100
-	45	00101101	E	69	01000101	]	93	01011101	u	117	01110101
.	46	00101110	F	70	01000110	^	94	01011110	v	118	01110110
/	47	00101111	G	71	01000111	_	95	01011111	w	119	01110111
0	48	00110000	H	72	01001000	`	96	01100000	x	120	01111000
1	49	00110001	I	73	01001001	a	97	01100001	y	121	01111001
2	50	00110010	J	74	01001010	b	98	01100010	z	122	01111010
3	51	00110011	K	75	01001011	c	99	01100011	{	123	01111011
4	52	00110100	L	76	01001100	d	100	01100100		124	01111100
5	53	00110101	M	77	01001101	e	101	01100101	}	125	01111101
6	54	00110110	N	78	01001110	f	102	01100110	~	126	01111110
7	55	00110111	O	79	01001111	g	103	01100111	□	127	01111111

# Таблица расширенного кода ASCII

символ	10-В код	2-В код									
Ъ	128	10000000		160	10100000	А	192	11000000	а	224	11100000
Г	129	10000001	Ў	161	10100001	Б	193	11000001	б	225	11100001
,	130	10000010	ў	162	10100010	В	194	11000010	в	226	11100010
г	131	10000011	Ј	163	10100011	Г	195	11000011	г	227	11100011
..	132	10000100	о	164	10100100	Д	196	11000100	д	228	11100100
...	133	10000101	Г	165	10100101	Е	197	11000101	е	229	11100101
+	134	10000110	!	166	10100110	Ж	198	11000110	ж	230	11100110
±	135	10000111	§	167	10100111	З	199	11000111	з	231	11100111
€	136	10001000	Е	168	10101000	И	200	11001000	и	232	11101000
‰	137	10001001	©	169	10101001	Й	201	11001001	й	233	11101001
Љ	138	10001010	€	170	10101010	К	202	11001010	к	234	11101010
<	139	10001011	«	171	10101011	Л	203	11001011	л	235	11101011
Њ	140	10001100	~	172	10101100	М	204	11001100	м	236	11101100
К	141	10001101	-	173	10101101	Н	205	11001101	н	237	11101101
Ђ	142	10001110	®	174	10101110	О	206	11001110	о	238	11101110
Џ	143	10001111	Ў	175	10101111	П	207	11001111	п	239	11101111
ђ	144	10010000	°	176	10110000	Р	208	11010000	р	240	11110000
‘	145	10010001	±	177	10110001	С	209	11010001	с	241	11110001
’	146	10010010	!	178	10110010	Т	210	11010010	т	242	11110010
“	147	10010011	і	179	10110011	У	211	11010011	у	243	11110011
”	148	10010100	г	180	10110100	Ф	212	11010100	ф	244	11110100
•	149	10010101	μ	181	10110101	Х	213	11010101	х	245	11110101
—	150	10010110	¶	182	10110110	Ц	214	11010110	ц	246	11110110
—	151	10010111	·	183	10110111	Ч	215	11010111	ч	247	11110111
□	152	10011000	ë	184	10111000	Ш	216	11011000	ш	248	11111000
™	153	10011001	№	185	10111001	Щ	217	11011001	щ	249	11111001
љ	154	10011010	€	186	10111010	Ъ	218	11011010	ъ	250	11111010
>	155	10011011	»	187	10111011	Ы	219	11011011	ы	251	11111011
њ	156	10011100	ј	188	10111100	Ь	220	11011100	ь	252	11111100
ќ	157	10011101	š	189	10111101	Э	221	11011101	э	253	11111101
ћ	158	10011110	š	190	10111110	Ю	222	11011110	ю	254	11111110
џ	159	10011111	ÿ	191	10111111	Я	223	11011111	я	255	11111111

# Избыточные коды

При записи и передаче данных часто используются **избыточные коды**, т. е. такие, которые за счет усложнения структуры позволяют повысить надежность передачи данных.

*Коды с обнаружением ошибок.*

Распространенным методом обнаружения ошибок является **контроль по четности**.

При записи байта информации в запоминающее устройство генерируется дополнительный контрольный бит, в который записывается «0», если это число четное, и «1», если оно нечетное. Если при чтении ранее записанного байта вновь получить контрольный бит и сравнить его с уже имеющимся, то можно судить о достоверности получаемой информации.

# Избыточные коды

При записи и передаче данных часто используются **избыточные коды**, т. е. такие, которые за счет усложнения структуры позволяют повысить надежность передачи данных.

# ошибок

Распространенным методом обнаружения ошибок является **контроль по четности.**

В этом случае при записи байта информации в запоминающее устройство генерируется **дополнительный контрольный бит**, в который записывается «0», если это число четное, и «1», если оно нечетное.

Если при чтении ранее записанного байта вновь получить контрольный бит и сравнить его с уже имеющимся, то можно судить о достоверности получаемой информации.

# Избыточные коды

Широко используется для обнаружения ошибок в блоках данных также код с циклическим контролем — *циклический избыточный код обнаружения ошибок (Cyclic Redundance Check — CRC)*.

Вычисляется **контрольная сумма** содержимого блока данных перед его передачей, включается в одно из полей блока, а затем повторно вычисляется после передачи. Несовпадение результатов свидетельствует об ошибке в передаваемом содержимом.

# Корректирующие коды

В ответственных приложениях, требующих повышенной надежности хранения информации, применяются более серьезные, чем контроль четности, методы обеспечения целостности данных. К ним относятся **корректирующие коды (Error Correction Code — ЕСС)**, позволяющие не только обнаруживать ошибки, но и **восстанавливать** искаженную информацию за счет ее избыточности.

# Системы счисления

Позиционные и непозиционные системы счисления.

**Системой счисления** называется совокупность приемов и правил для записи чисел цифровыми знаками.

# Непозиционные Системы счисления

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

Для их образования используют в основном операции сложения и вычитания.

Примером непозиционной системы счисления является римская система, использующая набор следующих символов: I, V, X, L, C, D, M и т. д.

# Позиционная система счисления

Позиционная система счисления – система, в которой значение символа определяется его положением в числе: один и тот же знак принимает различное значение.

Любая позиционная система характеризуется основанием.

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

**Основание позиционной системы счисления — количество ( $P$ ) различных цифр, используемых для изображения числа в позиционной системе счисления.**

**Значения цифр лежат в пределах от 0 до  $P - 1$ .**

$$**$N_{\max} = P^m - 1.$**$$

**$m$  – количество разрядов.**

# Пример

В двоичной системе ( $q=2$ )

$$11010.101_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3};$$

в) в шестнадцатиричной системе ( $q=16$ )

$$A3F.1CD_{16} = A \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0 + 1 \cdot 16^{-1} + C \cdot 16^{-2} + D \cdot 16^{-3}.$$

# Контрольные вопросы:

- 1. Что обеспечивает система счисления?**
- 2. Какая система счисления называется позиционной?**
- 3. Какая система счисления называется непозиционной?**
- 4. Какое равенство отождествляется с позиционной системой счисления?**
- 5. Приведите примеры позиционных и непозиционных систем счисления.**

# Недесятичная арифметика и её правила

## Двоичная арифметика

Арифметические операции во всех позиционных системах счисления выполняются по одним и тем же хорошо известным вам правилам.

# Сложение

Правила двоичного сложения:

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=10$$

# Пример

$$110_2$$

+

$$\underline{11}_2$$

$$1001_2$$

Проверим правильность вычислений сложением в десятичной системе счисления. Переведем двоичные числа в десятичную систему счисления и затем их сложим:

$$110_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6_{10};$$

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

$$6_{10} + 3_{10} = 9_{10}.$$

Теперь переведем результат двоичного сложения в десятичное число:

$$1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9_{10}$$

# Вычитание

Правила двоичного вычитания:

$$0 - 0 = 0$$

$$0 - 1 = 11$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

# Пример

$$110_2 - 11_2 :$$

$$110_2$$

$$\underline{-11_2}$$

$$11_2$$

# Умножение

Правила двоичного умножения:

$$0 * 0 = 0$$

$$0 * 1 = 0$$

$$1 * 0 = 0$$

$$1 * 1 = 1$$

# Пример

$110_2$

x 11<sub>2</sub>

110

+ 110

10010<sub>2</sub>

# Деление

**Деление.** Операция деления выполняется по алгоритму, подобному алгоритму выполнения операции деления в десятичной системе счисления. В качестве примера произведем деление двоичного числа  $110_2$  и  $11_2$ :

$$\begin{array}{r} 110_2 \quad | \quad \underline{\underline{11_2}} \\ - \quad | \quad 10_2 \\ \hline \underline{11} \\ 0 \end{array}$$

# Контрольные вопросы:

- 1. Какая арифметика называется недесятичной?**
- 2. Как производится сложение, вычитание, умножение, и деление в двоичной системе счисления?**
- 3. Как производится сложение, вычитание, умножение, и деление в восьмеричной системе счисления?**

# Представление чисел в ЭВМ

# Представление чисел в ЭВМ

В ЭВМ применяются две формы представления чисел:

- **естественная форма** или форма с фиксированной запятой (точкой)

и

- **нормальная форма** или форма с плавающей запятой.

# Форма с фиксированной запятой

Представление целых чисел

```
graph TD; A[Представление целых чисел] --> B[Положительные числа]; A --> C[Отрицательные числа];
```

Положительные числа

Отрицательные числа

# Фиксированная запятая (точка)

В форме представления с *фиксированной запятой (точкой)* числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

В современных компьютерах естественная форма представления используется как вспомогательная и **только для целых чисел.**

# ТОЧКОЙ

## (Floating point).

# Естественная форма

Целочисленные типы данных отлично подходят для работы с целыми числами, но часто приходится работать и с дробными числами.

И тут на помощь приходит тип данных с плавающей точкой (**floating point**).

Еще его называют тип данных с плавающей запятой.

Например: 4320.0, -3.33 или 0.01226.

**Почему точка «плавающая»?**

Дело в том, что точка или запятая может двигаться («плавать») между цифрами, разделяя таким образом целую и дробную части в значении.

# Числа с плавающей точкой

Экспоненциальная запись —  
представление действительных чисел в  
виде мантиссы и порядка.

$$N = M * n^p$$

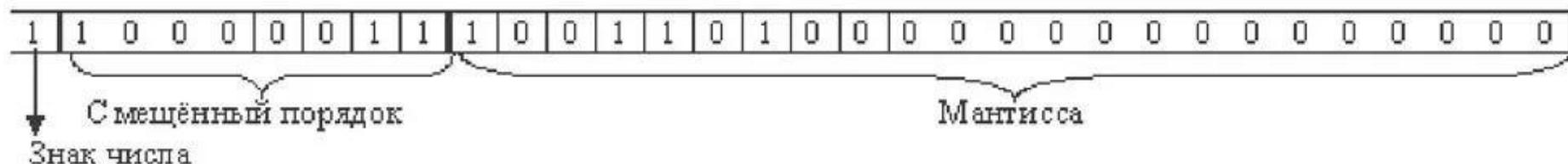
**N** — записываемое число;

**M** — мантисса;

**n** — основание;

**p** (целое) — порядок;

# Представление чисел с плавающей точкой в ЭВМ



Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.

Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего до наибольшего числа

**Двоично-десятичная система** счисления получила большое распространение в современных компьютерах ввиду легкости перевода в десятичную систему и обратно.

Используется там, где важно удобство работы пользователя. В двоично-десятичной системе счисления основанием системы счисления является число 10, но каждая десятичная цифра (0, 1, ..., 9) кодируется четырьмя **двоичными цифрами**.

Двоично-десятичная система не экономична с точки зрения реализации технического построения машины (примерно на 20 % увеличивается требуемое оборудование), но более удобна при подготовке задач и при программировании.

# Двоично-десятичная система счисления

Двоично-десятичный код				Десятичный код
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

**Таблица 1.8. Перевод цифр из двоичной системы счисления в восьмеричную, шестнадцатеричную и десятичную и наоборот**

Триада	Восьмеричная цифра	Тетрада	Шестнадцатеричная цифра	Десятичное число	Двоично-десятичная запись
000	0	0000	0	0	0000-0000
001	1	0001	1	1	0000-0001
010	2	0010	2	2	0000-0010
011	3	0011	3	3	0000-0011
100	4	0100	4	4	0000-0100
101	5	0101	5	5	0000-0101
110	6	0110	6	6	0000-0110
111	7	0111	7	7	0000-0111
		1000	8	8	0000-1000
		1001	9	9	0000-1001
		<b>1010*</b>	A	10	0001-0000
		<b>1011*</b>	B	11	0001-0001
		<b>1100*</b>	C	12	0001-0010
		<b>1101*</b>	D	13	0001-0011
		<b>1110*</b>	E	14	0001-0100
		<b>1111*</b>	F	15	0001-0101

# Контрольные вопросы

1. Какие формы представления чисел используются в ЭВМ?
2. Какие числа хранятся в каждой форме?
3. Какие числа обрабатываются быстрее?
4. Для чего применяется двоично-десятичный код?
5. Что такое дополнительный и обратный код? Для чего он используется?

# Прямой, обратный и дополнительный код

Входные и выходные данные представляются в форме, удобной для человека т.е. в 10-ой системе.

Для компьютера удобнее двоичная система.

Это объясняется тем, что **технически гораздо проще реализовать устройства** (например, запоминающий элемент) с двумя, а не с десятью устойчивыми состояниями (есть электрический ток — нет тока, намагничен — не намагничен и т.п.).



**Любые данные (числа, символы, графические и звуковые образы) в компьютере представляются в виде последовательностей из нулей и единиц.**

Обработку данных внутри компьютера можно воспринимать как преобразование слов из нулей и единиц по правилам, зафиксированным в микросхемах процессора.

# *Число с фиксированной запятой* формата слово со знаком:

Структурно запись числа  $-193(10) = -11000001(2)$  в разрядной сетке ПК выглядит следующим образом.

-193	Знак числа	Величина числа													
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
прямой	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
обратный	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1
Дополнительный	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1

# Целые числа

Для представления чисел в ЭВМ обычно используют битовые наборы — последовательности нулей и единиц фиксированной длины.

Организовать обработку наборов фиксированной длины технически легче, чем наборов переменной длины.

# Числа со знаком

Для представления знаковых целых чисел используются три способа:

- 1) прямой код;
- 2) обратный код;
- 3) дополнительный код.

Использование дополнительного и обратного кодов упрощает определение знака результата операции, вычитание есть сложение кодов, облегчено определение переполнения разрядной сетки

Структурно запись числа  $-193(10) = -11000001(2)$  в разрядной сетке ПК выглядит следующим образом.

-193	Знак числа	Величина числа														
№ разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
прямой	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

# Числа

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — цифровая часть содержит двоичную запись числа, в знаковом разряде содержится 0 (ноль)

Например, для  $k = 8$ :

Число  $1_{10} = 1_2$

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

Знак числа «+»

Число  $127_{10} = 1111111_2$

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

Знак числа «+»

$k$  - Количество разрядов

# Обратный код

## отрицательных чисел

Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями.

- Пример (Количество разрядов -  $k = 8$ ):

Число  $-1$

Код модуля числа:

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

Обратный код числа:

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

Знак числа «-»

Число  $-127$

Код модуля числа:

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

Обратный код числа:

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

Знак числа «-»

# Дополнительный код отрицательных чисел

Дополнительный код — наиболее распространённый способ представления отрицательных целых чисел в компьютерах.

Позволяет заменить операцию вычитания на операцию сложения и сделать операции сложения и вычитания одинаковыми для знаковых и беззнаковых чисел, чем упрощает архитектуру ЭВМ.



**Дополнительный код отрицательного числа получается инвертированием модуля двоичного числа и прибавлением к инверсии единицы, либо вычитанием числа из нуля.**

# Пример

Двоичное 8-ми разрядное число с отрицательным знаком:  $X = -01011101$

Получаем прямой код: минус - знак числа записывается в виде 1, коды числа записываются без изменения:

**$X_{\text{пр.}} = 1.01011101.$**

Для преобразования прямого кода двоичного отрицательного числа в обратный код знаковый разряд оставить без изменения, а в остальных разрядах нули заменить на единицы, а единицы на нули:

**$X_{\text{обр.}} = 1.10100010.$**

Для получения дополнительного кода необходимо к младшему разряду числа в обратном коде добавить единицу:

**$X_{\text{доп.}} = 1.10100011.$**

Для положительных двоичных чисел (в знаковом разряде записывается 0):  $X_{\text{пр.}} = X_{\text{обр.}} = X_{\text{доп.}}$

# Представление вещественных чисел

Вещественными числами называются числа, имеющие дробную часть.

Для хранения дробных чисел используется формат чисел **с плавающей запятой**

# Тип данных с плавающей точкой. Floating point.

Формат для хранения вещественных чисел называют тип данных с плавающей запятой (**floating point**).

Например: 4320.0, -3.33 или 0.01226.

**Почему запятая «плавающая»?**

Дело в том, что точка или запятая может двигаться («плавать») между цифрами, разделяя таким образом целую и дробную части в значении.

# Числа с плавающей точкой

Экспоненциальная запись —  
представление действительных чисел в  
виде мантиссы и порядка.

$$N = M * n^p$$

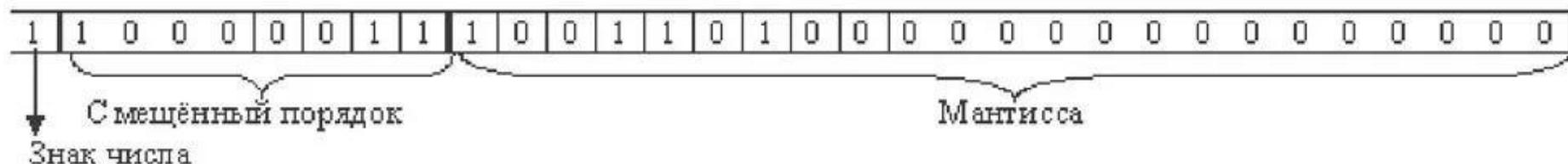
**N** — записываемое число;

**M** — мантисса;

**n** — основание;

**p** (целое) — порядок;

# Представление чисел с плавающей точкой в ЭВМ



Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.

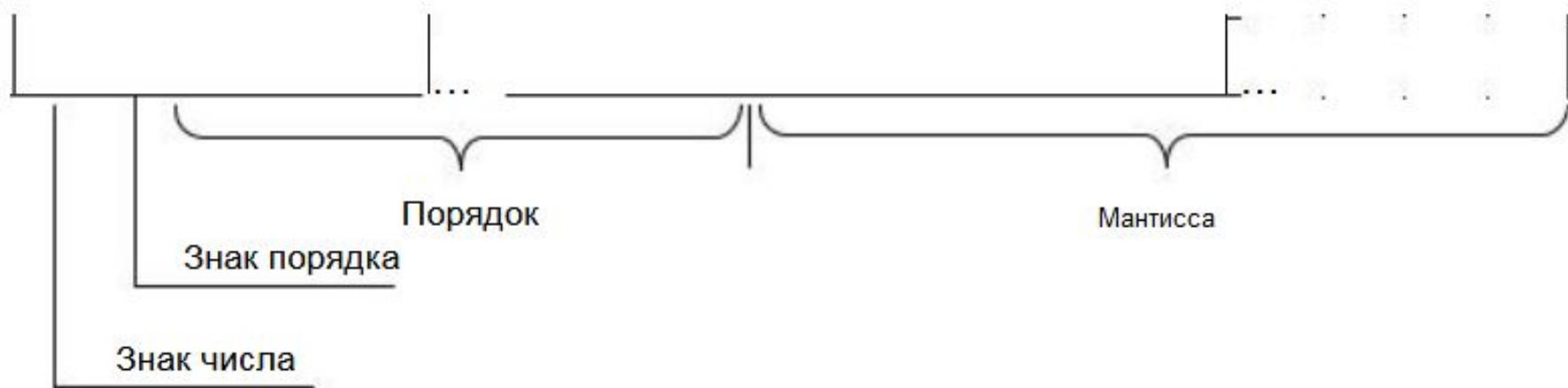
Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего до наибольшего числа

**Использование представления чисел с плавающей точкой существенно усложняет схему арифметико-логического устройства.**

**Для работы с числами с плавающей запятой выделена специальная часть процессора – математический сопроцессор**

# Представление в виде набора битов

Числа с плавающей точкой представляются в виде битовых наборов, в которых отводятся разряды для мантиссы, порядка, знака числа и знака порядка:





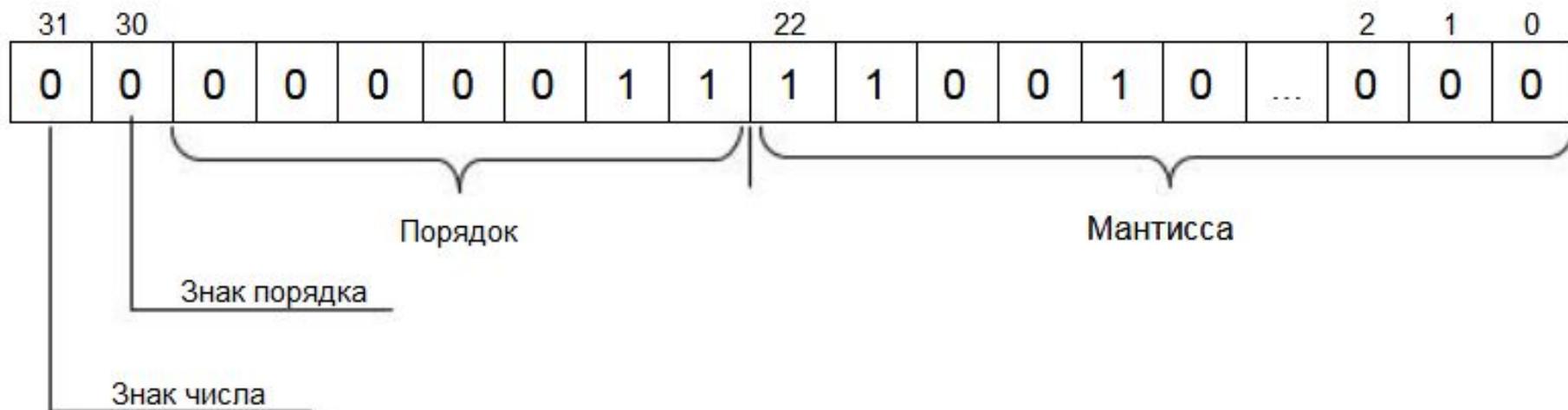
**Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.**

**Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.**

# Пример

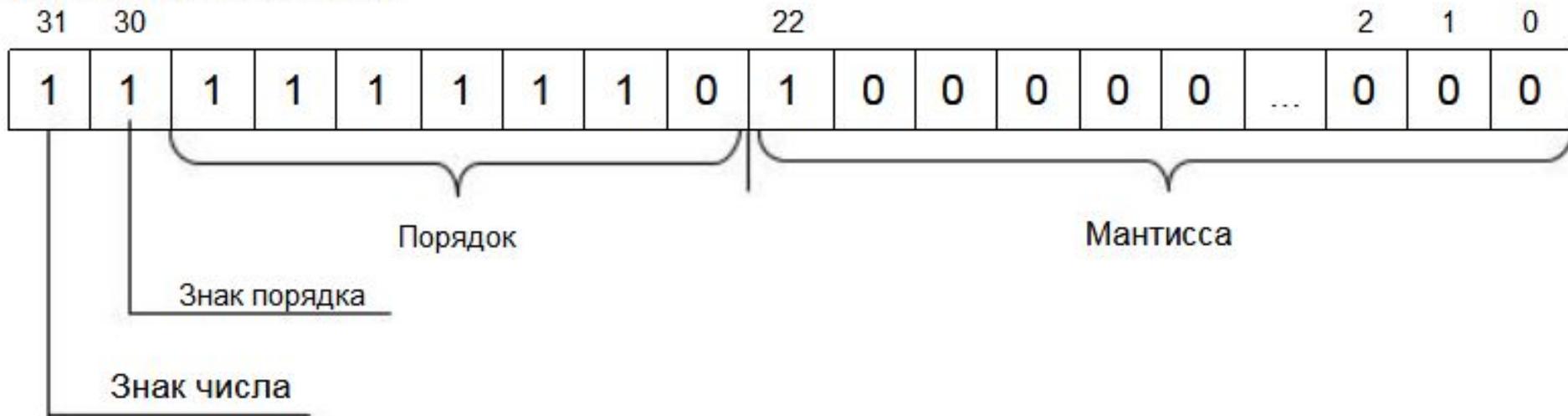
- Пример ( четырехбайтовое число формате с семью разрядами для записи порядка).

$$\text{Число } 6,25_{10} = 110,01_{(2)} = 0,11001_{(2)} \cdot 2^{11}$$



# Пример (отрицательное число)

Число  $-0,125_{10} = -0,001_2 = -0,1_2 \cdot 2^{-10}_2$  (отрицательный порядок записан в дополнительном коде):



# Представление вещественных чисел

Вещественными числами называются числа, имеющие дробную часть.

Для хранения дробных чисел используется формат чисел **с плавающей запятой**

# Тип данных с плавающей точкой. Floating point.

Формат для хранения вещественных чисел называют тип данных с плавающей запятой (**floating point**).

Например: 4320.0, -3.33 или 0.01226.

**Почему запятая «плавающая»?**

Дело в том, что точка или запятая может двигаться («плавать») между цифрами, разделяя таким образом целую и дробную части в значении.

# Числа с плавающей точкой

Экспоненциальная запись —  
представление действительных чисел в  
виде мантиссы и порядка.

$$N = M * n^p$$

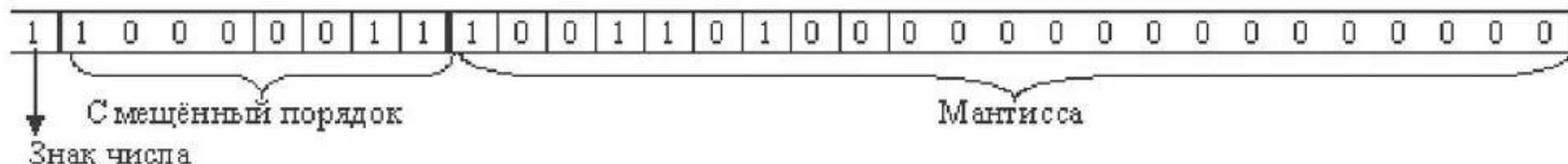
**N** — записываемое число;

**M** — мантисса;

**n** — основание;

**p** (целое) — порядок;

# Представление чисел с плавающей точкой в ЭВМ



Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.

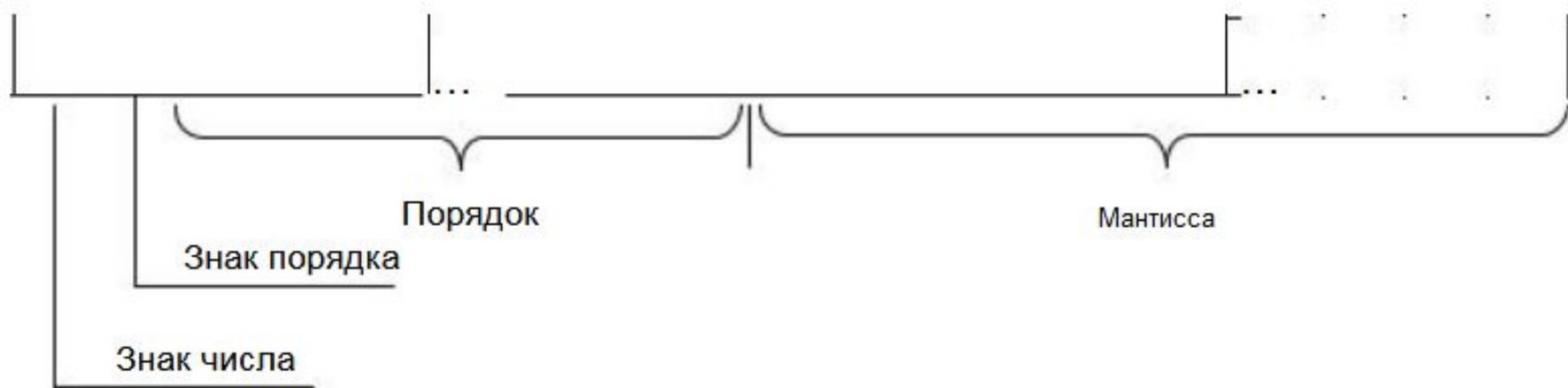
Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего до наибольшего числа

**Использование представления чисел с плавающей точкой существенно усложняет схему арифметико-логического устройства.**

**Для работы с числами с плавающей запятой выделена специальная часть процессора – математический сопроцессор**

# Представление в виде набора битов

Числа с плавающей точкой представляются в виде битовых наборов, в которых отводятся разряды для мантиссы, порядка, знака числа и знака порядка:





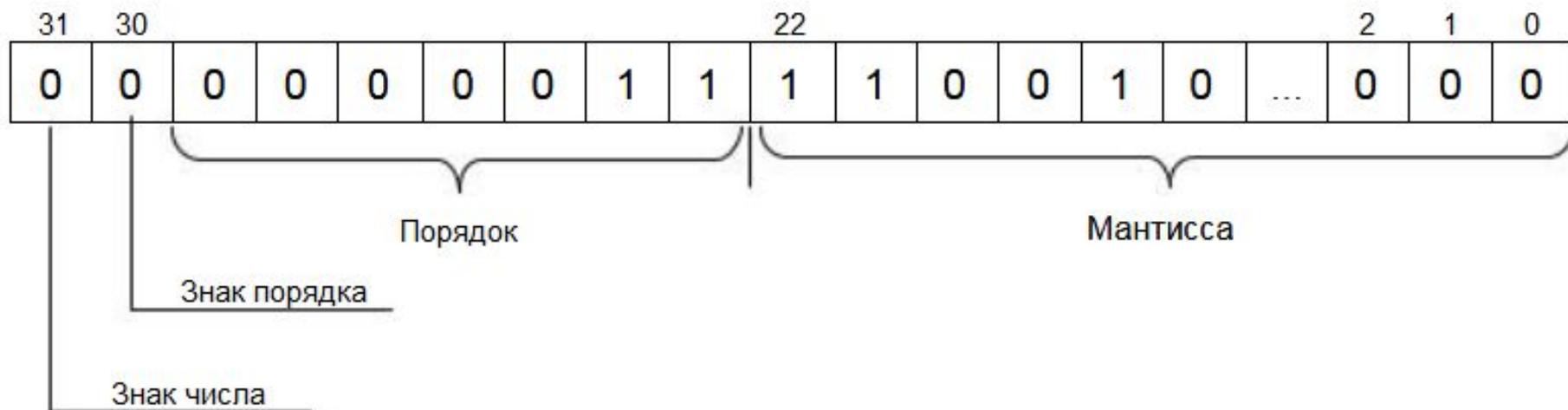
**Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.**

**Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.**

# Пример

- Примерах ( четырехбайтовом формате с семью разрядами для записи порядка).

$$\text{Число } 6,25_{10} = 110,01_{(2)} = 0,11001_{(2)} \cdot 2^{11}$$



# Пример (отрицательное число)

Число  $-0,125_{10} = -0,001_2 = -0,1_2 \cdot 2^{-10}_2$  (отрицательный порядок записан в дополнительном коде):

