

В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

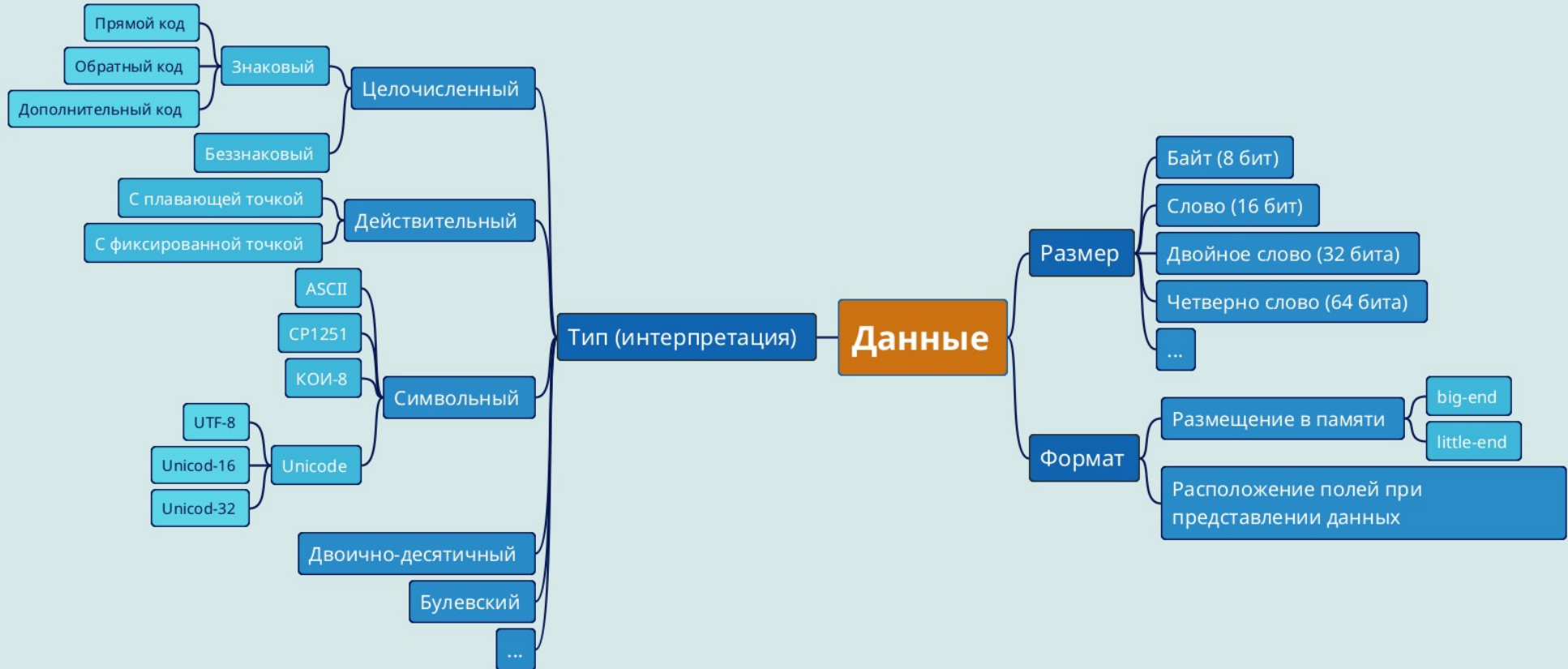
[illegible]

0000101001000010001010
 1000101001010101010110
 01001100010001010001
 01010000101001010011
 10011010010000001010
 01010010010010010100
 10010010101010101100
 10101010000100010011
 00101000100101001001

7981635743638426215419512897014101109712062804303795951955
712053292819162585258673257939844888829164476090575276957
78183520953666725212108357913458828091444200651033467110314
761813510560894099898999238735288231635507647918538923621
9461635282709819430992448899571282898950232332679297120844
798952266878312352658213149585872945341893939664262434
5227211660396615573092541105878537634668206531089862691862
204333489503661367668753249916680396267977877156084552963

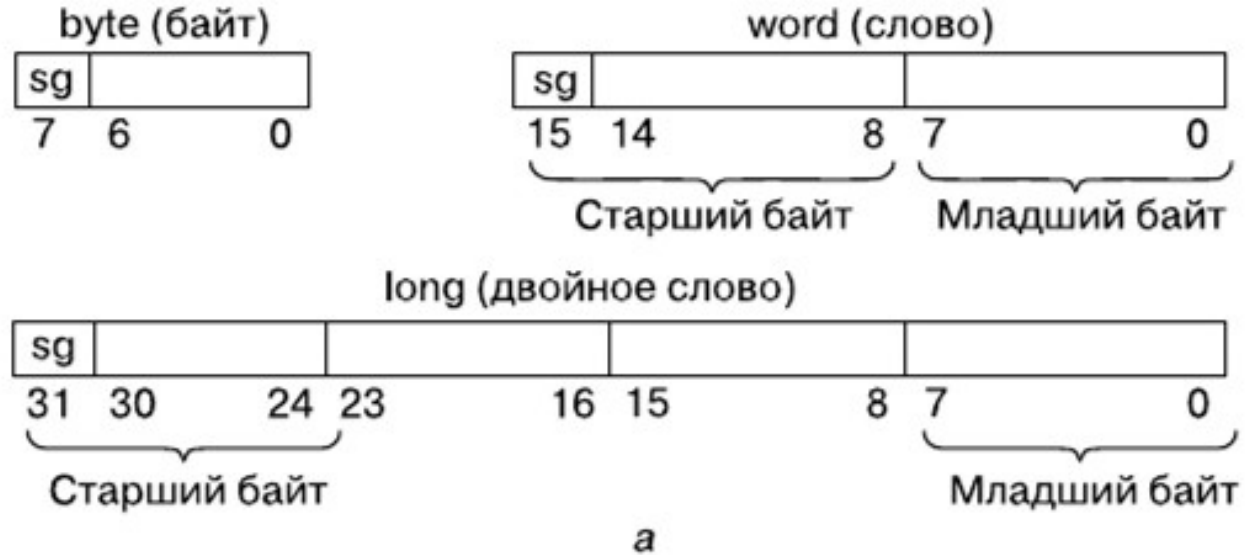
[illegible]

Характеристики данных

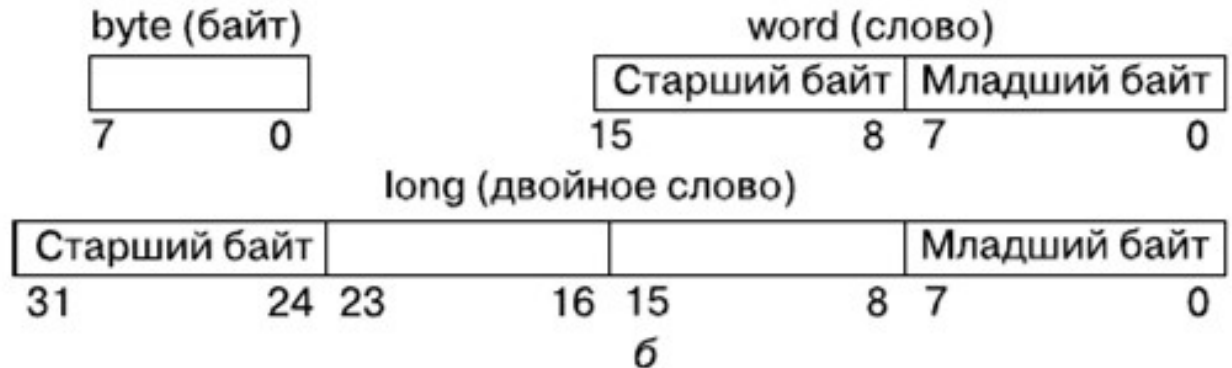


Форматы целочисленных данных

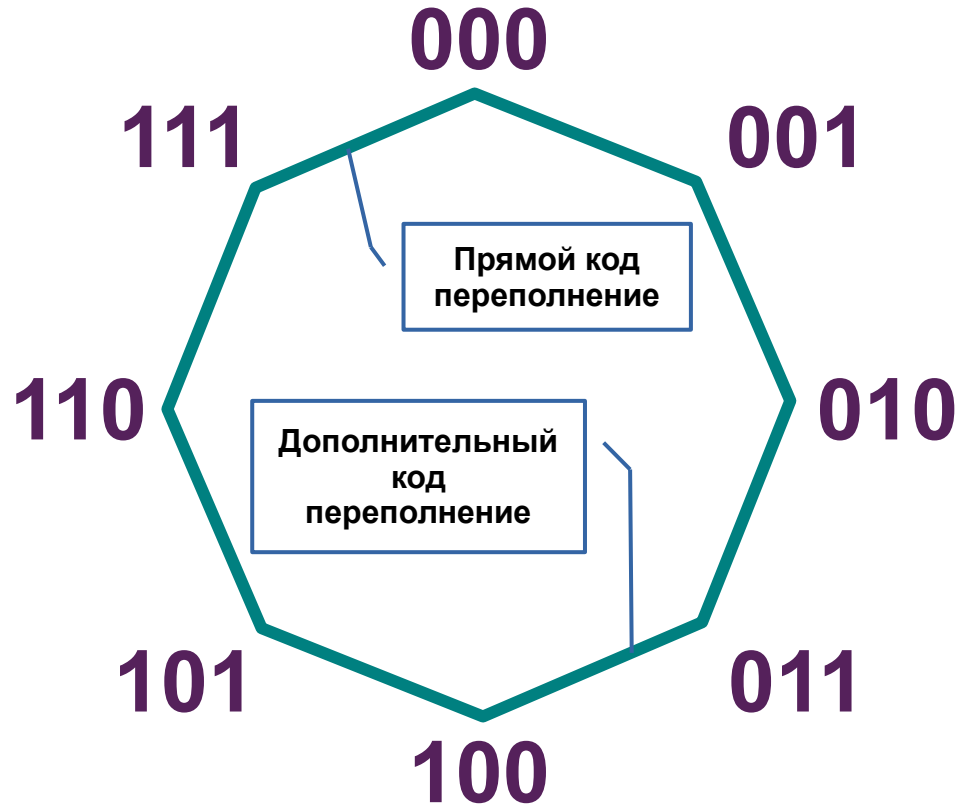
Знаковые числа:



Беззнаковые числа:



Представление целочисленных данных



Беззнаковая интерпретация

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

Прямой код

-3	-2	-1	-0	0	1	2	3
111	110	101	100	000	001	010	011

Обратный код

-3	-2	-1	-0	0	1	2	3
100	101	110	111	000	001	010	011

Дополнительный код

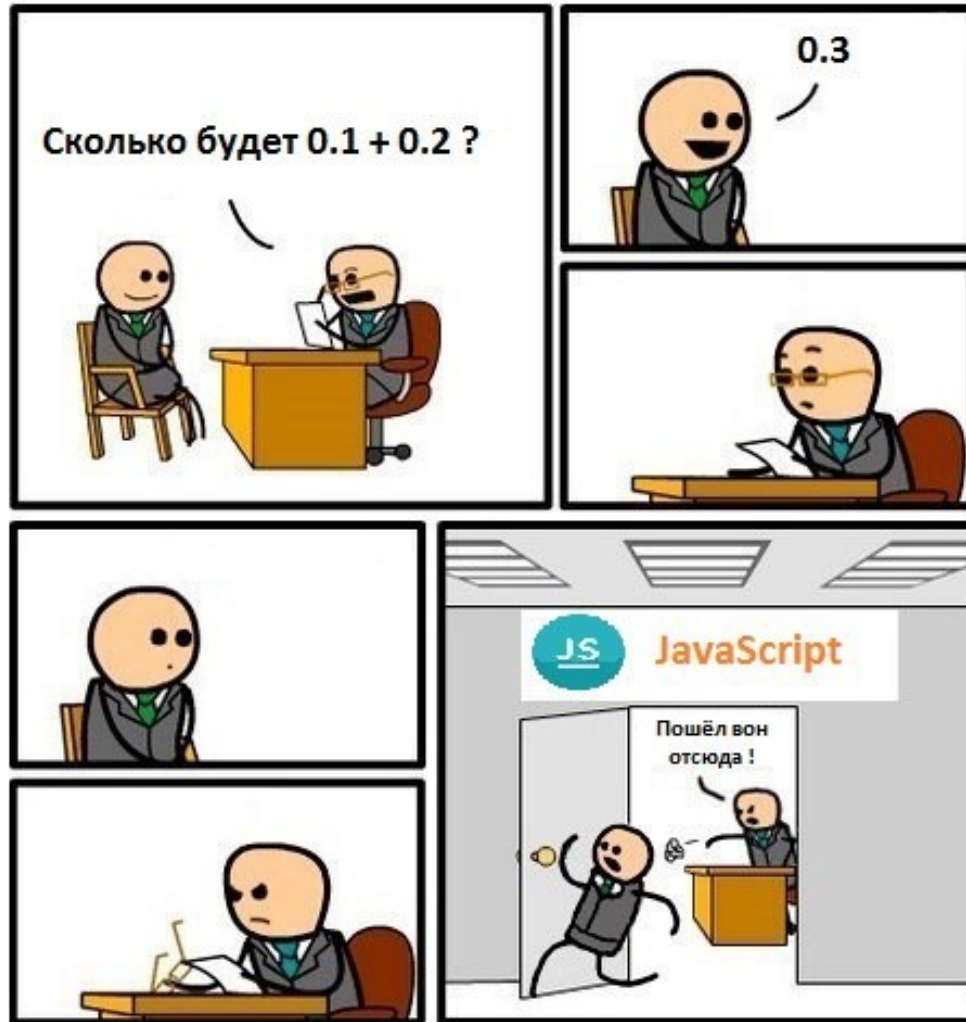
-4	-3	-2	-1	0	1	2	3
100	101	110	111	000	001	010	011



Представление символьных данных беззнаковыми числами

Символ	10й код	2й код	Символ	10й код	2й код	Символ	10й код	2й код	Символ	10й код	2й 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

Числа с плавающей точкой



Числа с плавающей точкой

```
~ python
```

```
Python 3.10.6 (main, Aug 3 2022, 17:39:45) [GCC 12.1.1 20220730] on linux  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> 0.1 + 0.2  
0.30000000000000004  
>>> quit()
```

```
~ node
```

```
Welcome to Node.js v16.16.0.  
Type ".help" for more information.
```

```
> 0.1 + 0.2  
0.30000000000000004  
> █
```

Числа с плавающей точкой

```
1 #include <stdio.h>
2
3 int main() {
4     printf("0.1 + 0.2 = %.17f\n", 0.1 + 0.2);
5     return 0;
6 }
```

```
~/Doc/E/Д/A/2022/pre/04/prog gcc 0.1+0.2.c
~/Doc/E/Д/A/2022/pre/04/prog ./a.out
0.1 + 0.2 = 0.3000000000000000004
```


Числа с плавающей точкой

```
1 #include <iostream>
2 #include <iomanip>
3
4 ▼ int main() {
5     std::cout << "0.1 + 0.2 = "
6         << std::fixed << std::setprecision(17)
7         << 0.1 + 0.2 << "\n";
8     return 0;
9 }
```



~/Doc/E/Д/A/2022/pre/04/prog

g++ 0.1+0.2.cpp
./a.out

0.1 + 0.2 = 0.30000000000000000004

Определения

Число с плавающей запятой (или число с плавающей точкой) — экспоненциальная форма представления вещественных (действительных) чисел, в которой число хранится в виде мантиссы и порядка (показателя степени). Имеет фиксированную относительную точность и изменяющуюся абсолютную. Наиболее часто представление утверждено в стандарте IEEE 754.

Определения

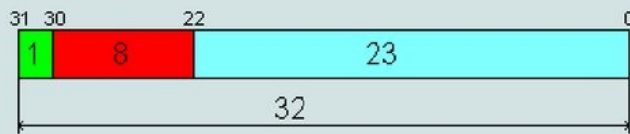
IEEE 754 (IEC 60559) — широко используемый стандарт IEEE, описывающий формат представления чисел с плавающей точкой. Используется в программных и аппаратных реализациях математических операций.

Стандарт описывает:

- формат чисел с плавающей точкой: мантиссу, экспоненту (показатель), знак числа;
- представление положительного и отрицательного нуля, положительной и отрицательной бесконечностей, а также нечисла (англ. Not-a-Number, NaN);
- методы, используемые для преобразования числа при выполнении математических операций;
- исключительные ситуации: деление на ноль, переполнение, потерю значимости, работа с денормализованными числами и другие;
- операции: арифметические и другие.

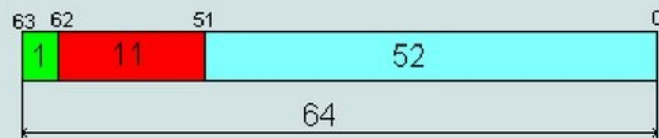
Стандарт IEEE 754

Для float (32 бит):



$$F = (-1)^S 2^{(E-127)} (1 + M/2^{23})$$

Для double (64 бит):



$$F = (-1)^S 2^{(E-1023)} (1 + M/2^{52})$$

Форматы чисел с плавающей точкой



Порядок со знаком

записан в смещённом коде

128	11111111
127	11111110
...	
2	10000001
1	10000000
0	01111111
-1	01111110
-2	01111101
...	
-126	00000001
-127	00000000

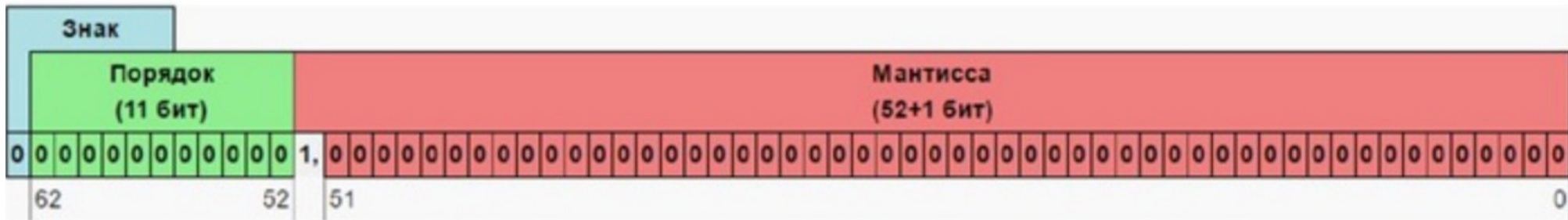
Из **мантиссы** записываются только 23 цифры дробной части (целая часть числа всегда равна 1, её хранить незачем!)

Знак числа: 0 – плюс, 1 – минус

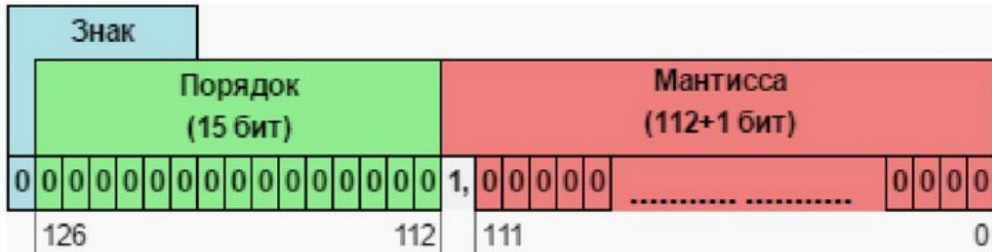
Максимальное число

$$2^{128} = 3,4028234 \times 10^{38}$$

Форматы чисел с плавающей точкой



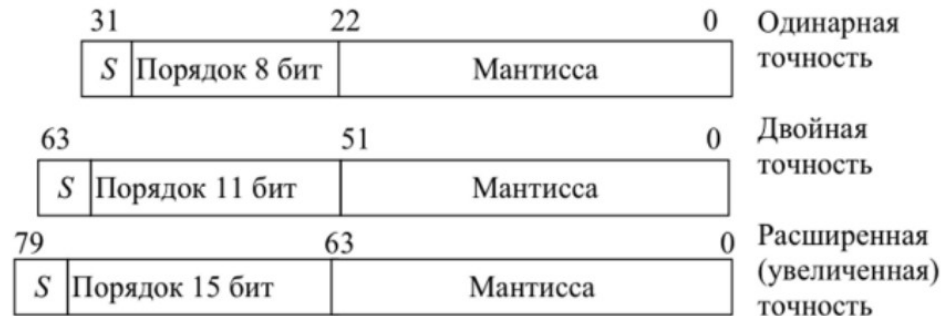
Сдвиг порядка = 1023



Сдвиг порядка = 16383



Сдвиг порядка = 15



Форматы Intel 8086

Форматы чисел с плавающей точкой

1. Нормализованное представление



2. Ненормализованное представление



3a. Бесконечность



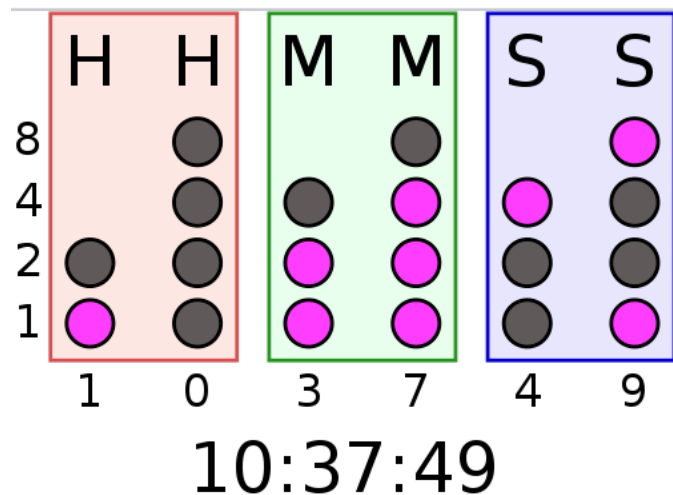
3b. NaN (не число)



<https://www.lua.org/>



Двоично-десятичное представление

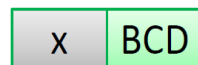


$$\begin{array}{r}
 \begin{array}{cc} * & ** \end{array} \\
 \begin{array}{cccc} 0011 & 1001 & 0010 & 0111 \\ + & 0100 & 1000 & 0101 & 0110 \end{array} \\
 \hline
 = & 1000 & 0001 & 0111 & 1101 & - \text{Двоичная сумма} \\
 + & & 0110 & & 0110 & - \text{Коррекция} \\
 \hline
 1000 & 0111 & 1000 & 0011
 \end{array}$$

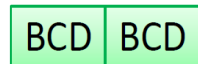
'*' — тетрада, из которой был перенос в старшую тетраду

'**' — тетрада с запрещённой комбинацией битов

Двоично-десятичный формат (Binary Coded Decimal - BCD) → → →



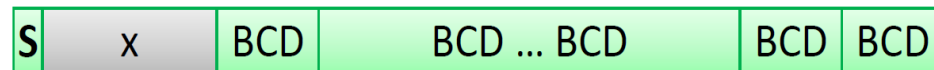
7 4 3 0



7 4 3 0

BCD Integer - один байт, содержит десятичную цифру

Packed BCD Integer - один байт, содержит две десятичные цифры



79 78 72 71 68 67

8 7 4 3 0

80-bit Packed BCD Integer - десять байтов (18 цифр плюс знак в старшем бите старшего байта), S - знак числа

Используемые источники

1. Таненбаум Э. Архитектура компьютера. 6-е изд. – СПб.: Изд. Питер, 2017. – 816 с.
2. Гагарина Л. Г., Кононова А. И. Архитектура вычислительных систем и Ассемблер с приложением методических указаний к лабораторным работам. Учебное пособие. — М.: СОЛОН-Пресс, 2019. - 368 с.

Википедия

Стандарт IEEE 754-2008: https://ru.wikipedia.org/wiki/IEEE_754-2008

Число с плавающей запятой: https://ru.wikipedia.org/wiki/Число_с_плавающей_запятой

Двоично-десятичный код: https://ru.wikipedia.org/wiki/Двоично-десятичный_код

Интернет

Что нужно знать про арифметику с плавающей запятой: <https://habr.com/ru/post/112953/>

Всё, точка, приплыли! Учимся работать с числами с плавающей точкой и разрабатываем альтернативу с фиксированной точностью десятичной дроби:

<https://habr.com/ru/company/xakep/blog/257897/>