

Компьютерная арифметика

- § 24. Особенности представления чисел в компьютере
- § 25. Хранение в памяти целых чисел
- § 26. Операции с целыми числами
- § 27. Хранение в памяти вещественных чисел
- § 28. Операции с вещественными числами

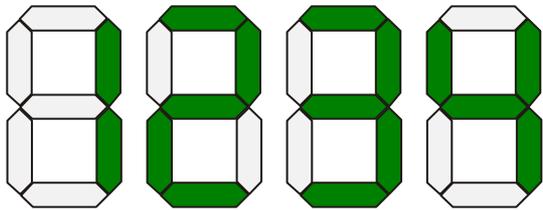
Компьютерная арифметика

§ 24. Особенности представления чисел в компьютере

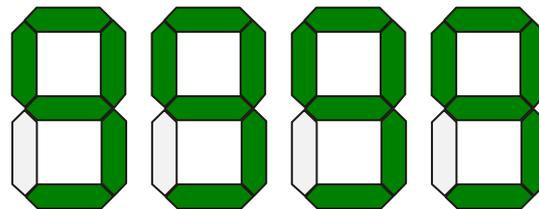
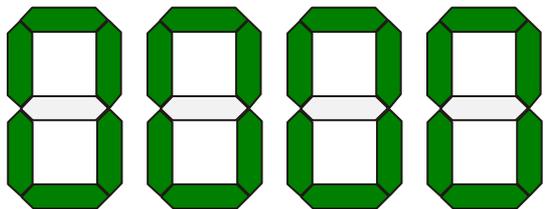
Предельные значения чисел

В математике *нет* предельных значений!

В компьютере – конечное число деталей, ограниченное количество разрядов.

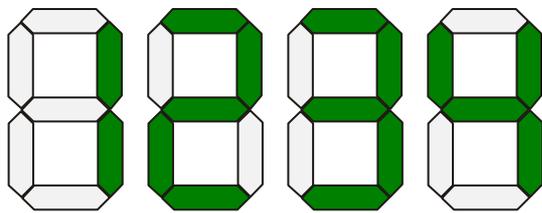


Какой диапазон?



~~10000?~~

Предельные значения чисел



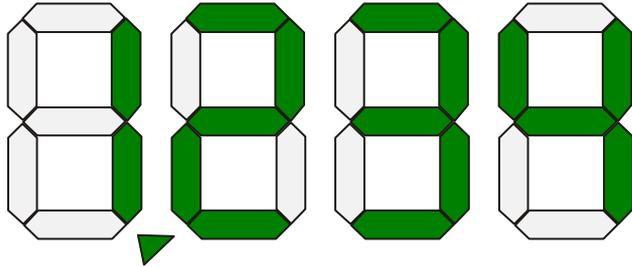
K разрядов

система счисления
с основанием B

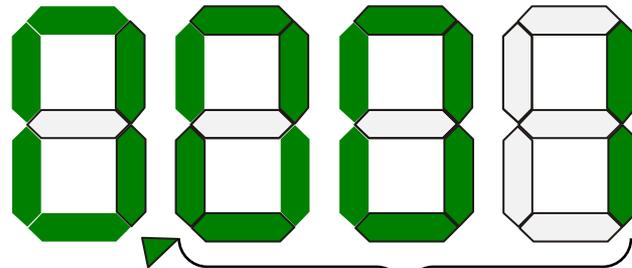
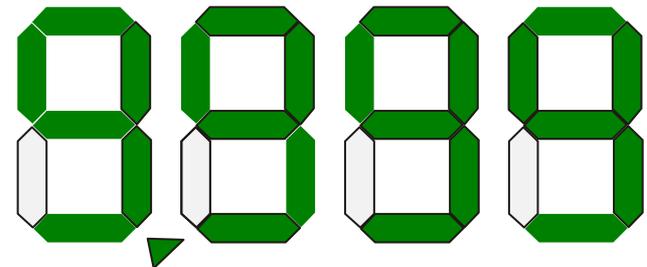
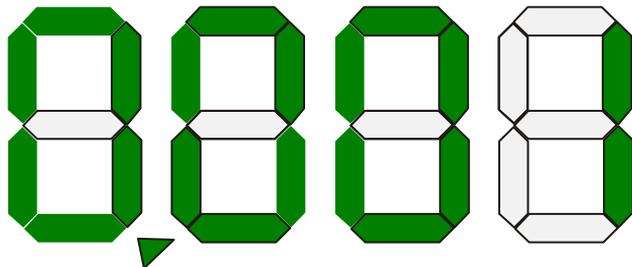
$$C_{\max} = B^K - 1$$

Переполнение разрядной сетки — это ситуация, когда число, которое требуется сохранить, не помещается в имеющемся количестве разрядов вычислительного устройства.

Вещественные числа



? Какой диапазон?

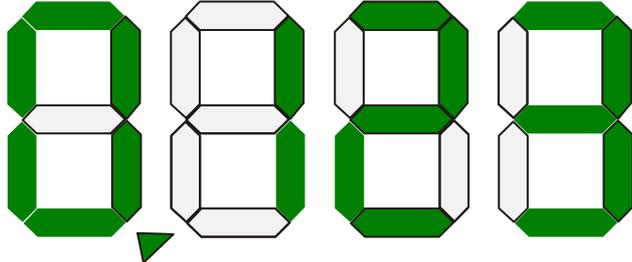


система счисления с
основанием B

F разрядов в
дробной части

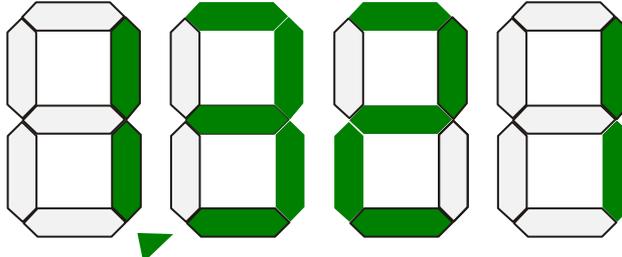
$$C_{\min} = B^{-F}$$

Неточность представления

0,123**4567** → 



Не все вещественные числа могут быть представлены в компьютере точно!

1,3211 →
1,3212 →
1,3214 → 

Сравнение вещественных чисел

$$X = 10^6, Y = 10^{-6}$$

хранится неточно!

$$X \cdot Y \approx 1$$

неточный результат!

$$0,3 \approx 0,1 + 0,1 + 0,1$$

! При сравнении вещественных чисел желательно избегать операции «равно»!

$$X \cdot Y \approx 1 \rightarrow |1 - X \cdot Y| \leq \varepsilon$$

допустимая погрешность (10^{-6})

$$X \approx Y \rightarrow |X - Y| \leq \varepsilon$$

Дискретность

1. Целые числа дискретны.
2. Вещественные числа **непрерывны**.
3. Компьютер работает только с **дискретными** данными.



Для хранения вещественных чисел
нужна **дискретизация!**

4. При дискретизации может происходить **потеря информации** (искажение данных).
5. Большинство **трудностей** связано с кодированием вещественных чисел.

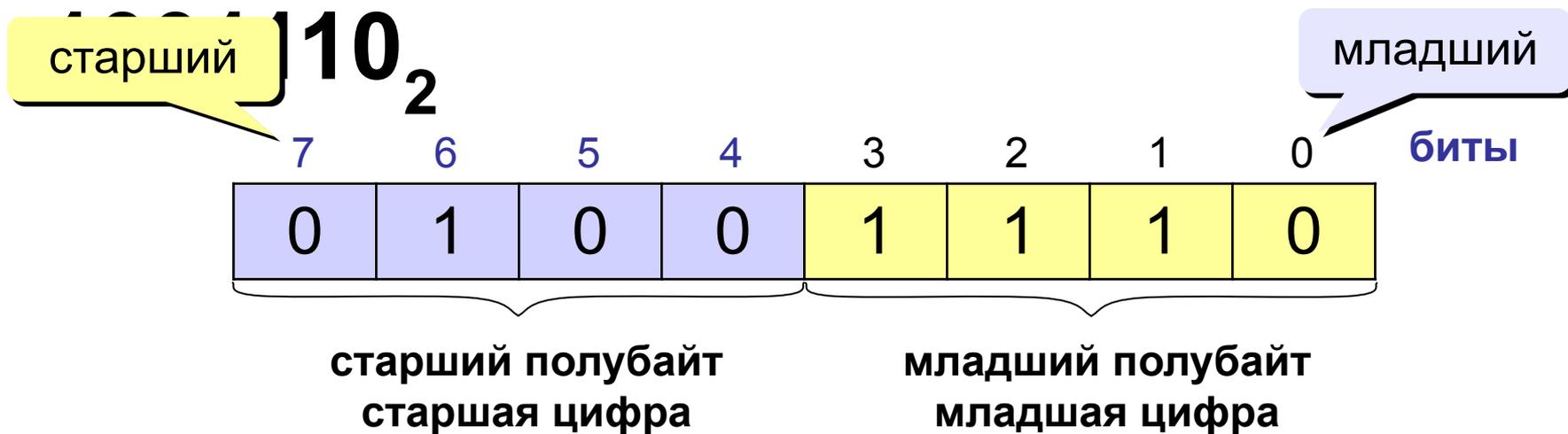
Компьютерная арифметика

§ 25. Хранение в памяти целых чисел

Целые числа без знака (*unsigned*)

Беззнаковые данные – не могут быть отрицательными.

78 =



4₁₆

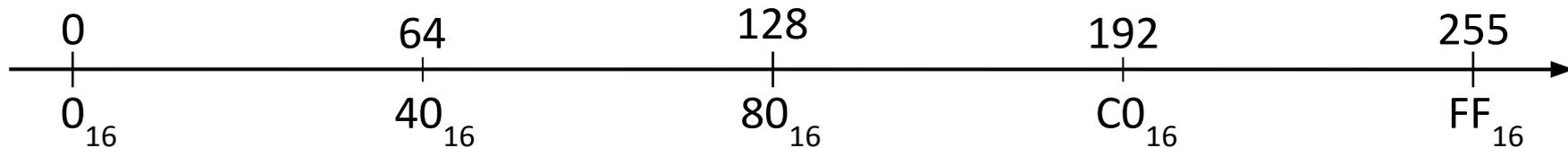
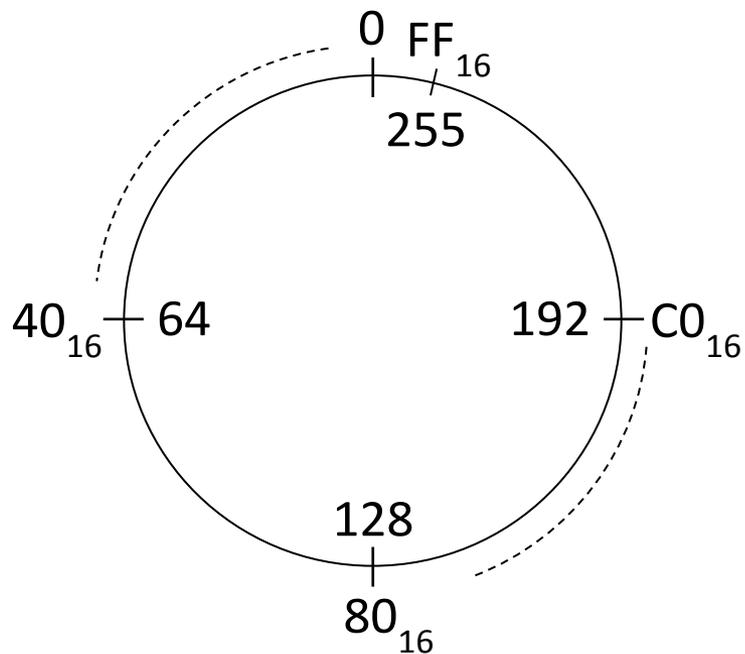
E₁₆

$$1001110_2 = 4E_{16} = 'N'$$

Целые числа без знака

X_{10}	0	1	...	127	128	...	255
X_{16}	00_{16}	01_{16}	...	$7F_{16}$	80_{16}	...	FF_{16}
X_2	$0000\ 0000_2$	$0000\ 0001_2$...	$0111\ 1111_2$	$1000\ 0000_2$...	$1111\ 1111_2$

$$\begin{array}{r}
 1111\ 1111 \\
 + 0000\ 0001 \\
 \hline
 1\ 0000\ 0000
 \end{array}$$



Целые числа без знака: диапазон

$$X_{\max} = 2^K - 1$$

K	X_{\min}	X_{\max}	типы данных
8	0	255	byte (Паскаль) unsigned char (C/C++)
16	0	65 535	word (Паскаль) unsigned short (C/C++)
32	0	4 294 967 295	cardinal (Delphi) unsigned int (C/C++)
64	0	18 446 744 073 709 551 615	unsigned long long (C/C++)

Целые числа со знаком



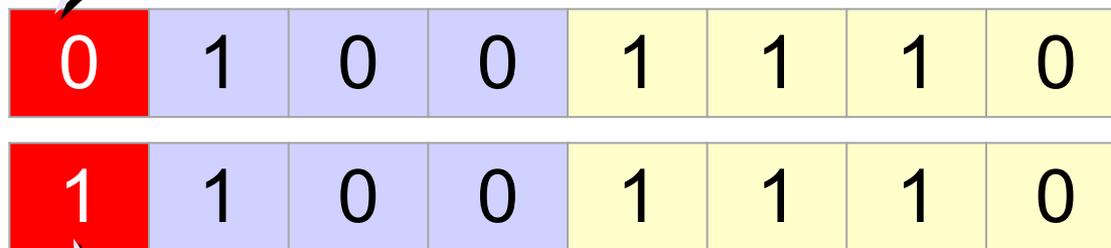
Сколько места требуется для хранения знака?

Старший (знаковый) бит числа определяет его знак. Если он равен 0, число положительное, если 1, то отрицательное.

Прямой код:

$$78 = 1001110_2$$

$$-78 = -1001110_2$$



≥ 0

< 0



операции с положительными и отрицательными числами выполняются по-разному!

Целые числа со знаком

Идея: «- 1» должно быть представлено так, чтобы при сложении с числом «1» получить 0.

 Как кодируется «-1»?

$$\begin{array}{r}
 1111 \ 1111 \quad -1 \rightarrow 255 \\
 + \ 0000 \ 0001 \quad 1 \\
 \hline
 1 \ 0000 \ 0000 \quad 256
 \end{array}$$

Для 8-битных чисел: код числа «-X» равен двоичному коду числа $256 - X$ (дополнение до 256).

 При K -битном кодировании *дополнительный* код числа «-X» равен двоичному коду числа $2^K - X$ (дополнение до 2^K).

Как построить дополнительный код?

Алгоритм А0: перевести число $2^k - X$ в двоичную систему счисления.

 для вычислений требуется $K+1$ разряд

Алгоритм А1:

- 1) перевести число X в двоичную систему счисления;
- 2) построить *обратный код*, выполнив *инверсию* всех битов (заменить 0 на 1 и наоборот);
- 3) к результату добавить 1.

$$\begin{array}{rcl}
 78 & = & 01001110_2 \\
 & & 10110001 \quad \leftarrow \text{инверсия} \\
 -78 & \rightarrow & 10110010 \quad +1
 \end{array}$$

Как построить дополнительный код?

Алгоритм А2:

- 1) перевести число $X-1$ в двоичную систему счисления;
- 2) выполнить инверсию всех битов.

$$78 - 1 = 77 = 01001101_2$$

$$-78 \rightarrow 10110010_2 \leftarrow \text{инверсия}$$

Алгоритм А3:

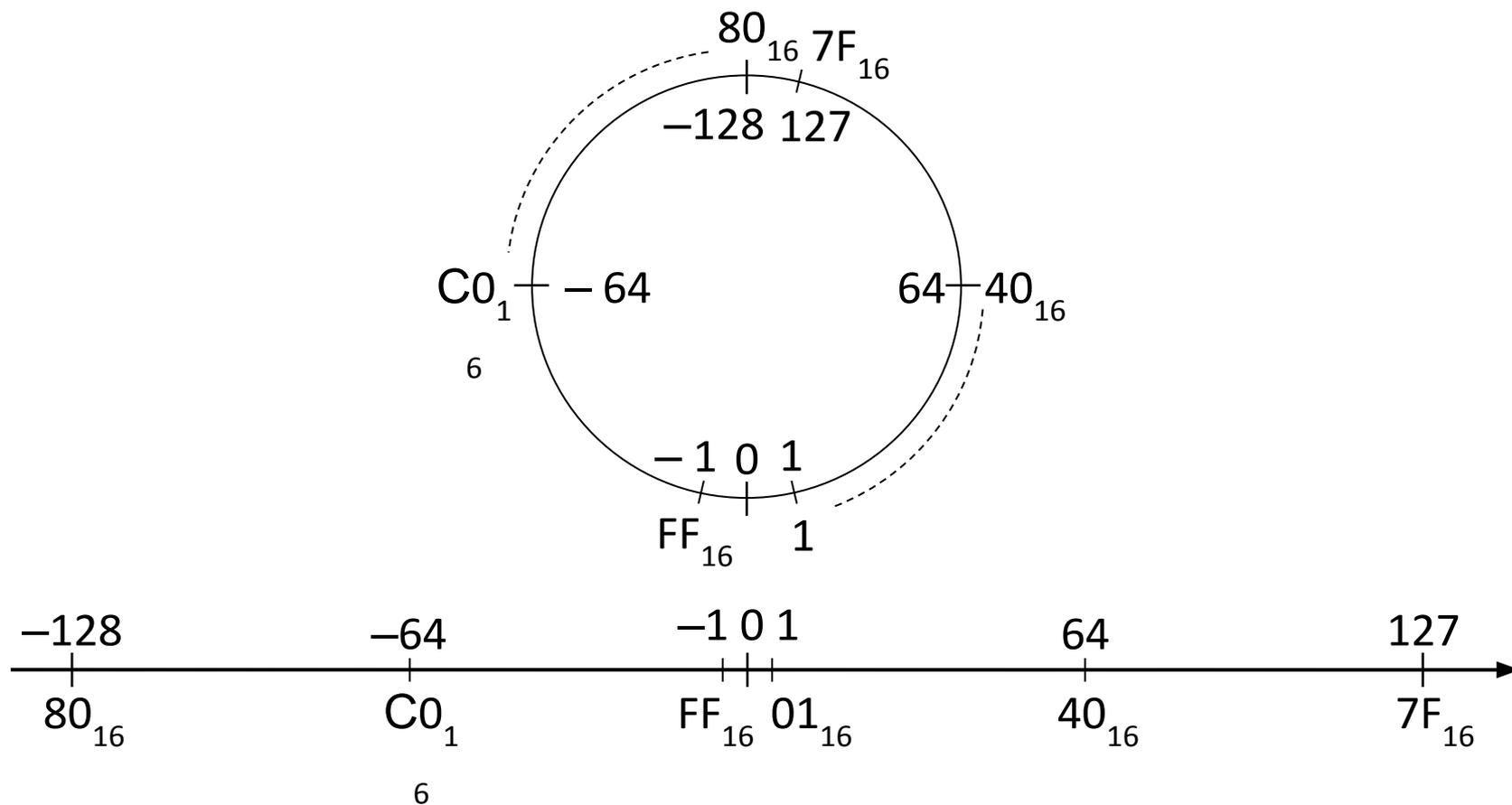
- 1) перевести число X в двоичную систему счисления;
- 2) выполнить инверсию всех старших битов числа, кроме младшей единицы и нулей после нее.

$$78 = 01001110_2$$

$$-78 \rightarrow 10110010_2 \leftarrow \text{инверсия}$$

Целые числа со знаком

X_{10}	-128	-127	...	-1	0	...	127
X_{16}	80_{16}	81_{16}	...	FF_{16}	00_{16}	...	$7F_{16}$
X_2	$1000\ 0000_2$	$1000\ 0001_2$...	$1111\ 1111_2$	$0000\ 0000_2$...	$0111\ 1111_2$



Целые числа со знаком: диапазон

$$X_{\min} = -2^{K-1}$$



$$X_{\max} = 2^{K-1} - 1$$

K	X_{\min}	X_{\max}	типы данных
8	-128	127	<code>shortInt</code> (Delphi) <code>char</code> (Cu)
16	-32 768	32 767	<code>smallInt</code> (Delphi) <code>short</code> (Cu)
32	-2 147 483 648	2 147 483 647	<code>integer</code> (Delphi) <code>int</code> (Cu)
64	-2^{63}	$2^{63} - 1$	<code>int64</code> (Delphi) <code>long long</code> (Cu)

