Представление целых чисел

Двоичная система счисления

Позиционные системы счисления:

- Десятичная:

$123_{10} = 1 * 10^2 + 2 * 10^1 + 3 * 10^0 = 1 * 100 + 2 * 10 + 3$

- Двоичная:

$$\begin{aligned}1111011_2 &= 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 \\ &= 1 * 64 + 1 * 32 + 1 * 16 + 1 * 8 + 0 * 4 + 1 * 2 + 1 = 123_{10}\end{aligned}$$

- Шестнадцатиричная:

$7B_{16} = 7 * 16^1 + B * 16^0 = 7 * 16 + 11 = 123_{10}$

Сколько чисел можно записать, используя N бит:

	Доступные числа	Сколько чисел
8 бит:	$0 \ldots 2^8 - 1 = 0 \ldots 255$	$2^8 = 256$
16 бит:	$0 \ldots 2^{16} - 1 = 0 \ldots 65535$	$2^{16} = 65536$
32 бита:	$0 \ldots 2^{32} - 1 = 0 \ldots 4294967295$	$2^{32} = 4294967296$
64 бита:	$0 \ldots 2^{64} - 1 = 0 \ldots 18446744073709551615$	$2^{64} = 18446744073709551616$

Байты

- Все операции по записи и обработке данных проводятся фрагментами, размер которых равен 1 байту = 8 бит
  - нельзя сделать файл размером 3 бита.
  - арифметические операции выполняются с числами размером 8, 16, 32 или 64 бита.

- Переполнение (1-байтные числа):

…	254	255	0	1	2	…	254	255	0	1	2	…

$$255 + 1 = 0$$
$$1 - 3 = 254$$

  - Арифметические операции выполняются «по модулю 256».

Отрицательные числа

Прямой код	• Signed magnitude representation • $-(2^{N-1}-1), \dots, -0, 0, \dots, (2^{N-1}-1)$
Обратный код	• One's complement • $-(2^{N-1}-1), \dots, -0, 0, \dots, (2^{N-1}-1)$
Дополнительный код	• Two's complement • $-2^{N-1}, \dots, 0, \dots, (2^{N-1}-1)$

Прямой код (Signed Magnitude Representation)

- Признаком отрицательного числа является старший бит.

Число	-127	...	-1	-0	0	1	...	127
Код	11111111_2	...	10000001_2	10000000_2	00000000_2	00000001_2	...	01111111_2
без знака	255	...	129	128	0	1	...	127

- Переполнение: $127 + 1 = -0$.

- Доступные числа: $(-2^{n-1}-1), \dots, -0, +0, \dots, +(2^{n-1}-1)$

8 бит: $-2^7 - 1 \dots 2^7 - 1$ $\quad -127 \dots 127$

16 бит: $-2^{15} - 1 \dots 2^{15} - 1$ $\quad -32767 \dots 32767$

32 бита: $-2^{31} - 1 \dots 2^{31} - 1$ $\quad -2147483647 \dots 2147483647$

64 бита: $-2^{63} - 1 \dots 2^{63} - 1$ $\quad -9223372036854775807 \dots 9223372036854775807$

Обратный код (One’s complement)

- Отрицательное число формируется путём вычисления побитового дополнения.

Число	-127	…	$-\mathbf{1}$	$-\mathbf{0}$	0	1	…	127
Код	10000000_2	…	11111110_2	11111111_2	00000000_2	00000001_2	…	01111111_2
без знака	128	…	254	255	0	1	…	127

- Переполнение: $127 + 1 = -127$.

- Доступные числа: $(-2^{n-1}-1), \dots, -0, +0, \dots, +(2^{n-1}-1)$

8 бит: $-2^7 - 1 \dots 2^7 - 1$ $-127 \dots 127$

16 бит: $-2^{15} - 1 \dots 2^{15} - 1$ $-32767 \dots 32767$

32 бита: $-2^{31} - 1 \dots 2^{31} - 1$ $-2147483647 \dots 2147483647$

64 бита: $-2^{63} - 1 \dots 2^{63} - 1$ $-9223372036854775807 \dots 9223372036854775807$

Дополнительный код (Two’s complement)

- Для представления отрицательных чисел инвертируем код положительного числа и добавляем 1.

Число	-128	…	-2	-1	0	1	…	127
Код	10000000_2	…	11111110_2	11111111_2	00000000_2	00000001_2	…	01111111_2
без знака	128	…	254	255	0	1	…	127

- Переполнение: $127 + 1 = -128$.

- Доступные числа: $(-2^{n-1}), \dots, 0, \dots, +(2^{n-1}-1)$

8 бит: $-2^7 \dots 2^7 - 1$ $-128 \dots 127$

16 бит: $-2^{15} \dots 2^{15} - 1$ $-32768 \dots 32767$

32 бита: $-2^{31} \dots 2^{31} - 1$ $-2147483648 \dots 2147483647$

64 бита: $-2^{63} \dots 2^{63} - 1$ $-9223372036854775808 \dots 9223372036854775807$

Переполнение при арифметических операциях

Алгоритм, применённый к двоичным строкам	Интерпретация битовых строк как беззнаковых чисел	Интерпретация битовых строк как чисел со знаком
1111 11 0111 0011 + 1101 0001 --------- 0100 0100	115_{10} + 209_{10} --- 68_{10} Результат неверный	115_{10} + -47_{10} --- 68_{10} Результат верный

Для беззнаковых чисел: Результат двоичного сложения верен, если при сложении не было переноса из старшего разряда.

Для чисел со знаком: Результат двоичного сложения верен, если при сложении перенос в старший разряд равен переносу из старшего разряда. При этом перенос может быть как 1 так и 0.

Переполнение при арифметических операциях

Битовое сложение	11111 111 0011 1111 1101 0101 ---------- 0001 0100	00000 011 1100 0001 0010 1011 ----------- 1110 1100	01111 100 0011 1111 0110 0100 ---------- 1010 0011	10000 000 1100 0001 1001 1100 ---------- 0101 1101
Интерпретация без знака	63_{10} + 213_{10} --- 20_{10}	213_{10} + 43_{10} --- 236_{10}	63_{10} + 100_{10} --- 163_{10}	213_{10} + 156_{10} --- 93_{10}
Интерпретация со знаком	63_{10} + -43_{10} --- 20_{10}	-63_{10} + 43_{10} --- -20_{10}	63_{10} + 100_{10} --- -93_{10}	-63_{10} + 100_{10} --- 93_{10}

Для беззнаковых чисел: Результат двоичного сложения верен, если при сложении не было переноса из старшего разряда.

Для чисел со знаком: Результат двоичного сложения верен, если при сложении перенос в старший разряд равен переносу из старшего разряда. При этом перенос может быть как 1 так и 0.

Двоично-десятичный код (Binary-coded decimal)

BCD – форма записи рациональных чисел, в которой для записи каждого разряда десятичного числа используют четырёхбитный двоичный код.

- Распакованный BCD
  - Для записи каждой цифры используется 1 байт:

$$91_{10} = 0000\ 1001 \quad 0000\ 0001$$

- Упакованный BCD
  - В одном байте записывают две цифры:

$$91_{10} = 1001\ 0001$$

Представление вещественных чисел по стандарту IEE 754

- Для представления вещественных чисел используют следующий формат $v = (-1)^s 1.b \times 2^e$:

	Одинарная точность	Двойная точность
Знак s	1 бит	1 бит
Мантисса b	8 бит	52 бита
Порядок e	7 бит	11 бит

- Есть коды для специальных значений: $+\infty, -\infty, NaN$
- Последствия:
  - точность представления маленьких по модулю чисел больше, чем больших;
  - не можем точно представлять большинство целых чисел;
  - не можем точно представлять бесконечные десятичные дроби;
  - результат выполнения операций зависит от их порядка, $a + b - a \neq b, \ldots$

Другие типы данных для чисел

- Специальные типы данных используются для работы:
  - целыми числами с большим (неограниченным) числом разрядов;
  - дробными числами ($\frac{456}{789}$);
  - десятичными дробями с фиксированным числом знаков после запятой (123.45).

- Для работы с такими числами часто используют специальные библиотеки.

- Некоторые новые языки программирования могут иметь поддержку таких чисел.