М. Е. Фиошин, А. А. Рессин, С. М. Юнусов

ИНФОРМАТИКА и ИКТ

ЧАСТЬ 1

ПРОФИЛЬНЫЙ УРОВЕНЬ

Под редакцией академика
Российской академии образования,
доктора педагогических наук,
профессора А. А. Кузнецова

Рекомендовано
Министерством образования и науки
Российской Федерации

10–11
КЛАССЫ

МОСКВА
ДРОФА
2·0·0·8

Прямой, обратный и дополнительный коды

В ЭВМ для представления чисел со знаком используются специальные коды — **прямой, обратный и дополнительный.** Идея использования этих кодов заключается в том, что сам код трактуется как число без знака, однако в самом коде предусматриваются две части: одна определяет знак числа, а вторая модуль числа. Под знак числа отводится специальный знаковый разряд (знаковый бит), который приписывается слева от модуля числа. Как правило, знаковым разрядом является крайний левый разряд в разрядной сетке. Значение нуль знакового бита определяет число как положительное, а единица как отрицательное.

Например, если за основу представления кода взят один байт, то для представления числа будет отведено 7 разрядов, а для записи кода знака один разряд.

0	1 1 1 1 0 0 1

Положительное число

1	1 0 0 1 0 0 1

Отрицательное число

В дальнейшем при записи положительных и отрицательных чисел знаковый разряд будет отделяться апострофом. Например, 0` 0011100, 1` 000101.

Положительные числа в прямом, обратном и дополнительном кодах записываются одинаково. При этом их изображение совпадает с записью самого числа и занесением в знаковый разряд 0. Например, +1001 записывается как 0\` 0001001.

Отрицательные числа в отличие от положительных записываются в прямом, обратном и дополнительном кодах по-разному.

Прямой код отрицательного числа совпадает по изображению с записью самого числа, при этом значение знакового разряда –1.

Пример.

При выделении для записи кода одного байта для числа –1101 прямой код этого числа будет записан так: 1\`0001101.

Обратный код отрицательного числа получается из прямого кода путем замены всех цифр в разрядах на противоположные — инверсия (1 на 0, 0 на 1), за исключением единицы в знаковом разряде.

Пример.

Для числа 1101 прямой код — 1\`0001101, а обратный — 1\`1110010.

Дополнительный код отрицательного числа образуется путем получения обратного кода и добавлением к младшему разряду единицы (с учетом переносов между разрядами).

Пример.

Для числа –1101:

Прямой код	Обратный код	Дополнительный код
1\`0001101	1\`1110010	1\`1110011

Последовательность перевода отрицательных чисел из прямого кода в обратный и дополнительный можно представить так:

$$N_{\text{прям.код}} \Rightarrow N_{\text{обрат.код}} = \overline{N_{\text{прям.код}}} \Rightarrow N_{\text{допл.код}} = N_{\text{обрат.код}} + 1.$$

Сложение чисел в обратном и дополнительном кодах

При сложении двоичных чисел положительные числа представляют в прямом коде, а отрицательные — в обратном или дополнительном.

> **!** При сложении чисел в обратном коде складываются все разряды, включая и разряды знака. Соответственно, если в знаковом разряде будет 0, то число положительное, если 1 — отрицательное. Однако если в знаковом разряде будет получена единица переноса, то она **прибавляется** к младшему разряду суммы кодов.

При сложении чисел в дополнительном коде также складываются все разряды, однако возникающая единица переноса в знаковом разряде **отбрасывается.**

> **!** Если в результате арифметических действий получено отрицательное число (код знака равен 1), его необходимо преобразовать в прямой код.

Для преобразования обратного кода в прямой все цифры в разрядах меняются на противоположные (инвертируются), а для преобразования дополнительного кода в обратный необходимо вычесть единицу.

Последовательность перевода отрицательных чисел из дополнительного кода в прямой можно представить так:

$$N_{\text{допл.код}} \Rightarrow N_{\text{обрат.код}} = N_{\text{допл.код}} - 1 \Rightarrow N_{\text{прям.код}} = \overline{N_{\text{обрат.код}}}$$

или

$$N_{\text{допл.код}} \Rightarrow N_{\text{прям.код}} = N_{\text{обрат.код}} + 1.$$

Рассмотрим примеры сложения чисел в обратном коде.

а) 7 + 5 = 12

$7_{10} \rightarrow 0`\ 0000111_2$

$5_{10} \rightarrow 0`\ 0000101_2$

$$\begin{array}{r} 0`\ 0000111_2 \\ +\ 0`\ 0000101_2 \\ \hline 0`\ 0001101_2 \end{array}$$

Ответ: $0`\ 0001101_2 = 12_{10}$

б) 8 + (–3) = 5

$8_{10} \rightarrow 0`\ 0001000_2$

$(-3_{10}) \rightarrow 1`\ 0000011_2 \rightarrow 1`1111100_{2\,\text{обр}}$

$$\begin{array}{r} 0`\ 0001000_2 \\ +\ 1`\ 0000011_{2\,\text{обр}} \\ \hline 10`\ 0000100_2 \\ \longrightarrow 1 \\ 0`\ 0000101_2 \end{array}$$

Ответ: $1`\ 0000101_2 = 5_{10}$

в) $6_{10} + (-11)_{10} = -5_{10}$

$6_{10} \to 0`\ 0000110_2$
$-11_2 \to 0`\ 0001011_2 \to 1`\ 1110100_{2\ обр}$

$$\begin{array}{r} 0`\ 0000110_2 \\ +\ 1`\ 1110100_2 \\ \hline 1`\ 1111010_{2\ обр} \end{array}$$

Ответ: $1`\ 0000101_2 = -5_{10}$

г) $(-10_{10}) + (-12_{10}) = -22_{10}$

$-10_{10} \to 1`\ 0001010_2 \to 1`1110101_{2\ обр}$
$-12_{10} \to 1`\ 0001100_2 \to 1`1110011_{2\ обр}$

$$\begin{array}{r} 1`\ 1110101_{2\ обр} \\ +\ 1`\ 1110011_{2\ обр} \\ \hline 11`\ 1101000_{2\ обр} \\ \longrightarrow 1 \\ \hline 1`\ 1101001_{2\ обр} \end{array}$$

Ответ: $1`\ 0010110_2 = -22_{10}$

Рассмотрим сложение тех же чисел, что и в предыдущих примерах, но только в дополнительных кодах.

а) $(-10)_{10} + (-12)_{10} = -22_{10}$

$-10_{10} \to 1`\ 0001010_2 \to 1`\ 1110101_{2\ обр} \to 1`\ 1110110_{2\ доп}$
$-12_{10} \to 1`\ 0001100_2 \to 1`\ 1110011_{2\ обр} \to 1`\ 1110100_{2\ доп}$

$$\begin{array}{r} 1`\ 1110110_{2\ доп} \\ +\ 1`\ 1110100_{2\ доп} \\ \hline 11`\ 1101010_{2\ доп} \end{array} \to 1`\ 0010101_{2\ обр} + 1`\ 0010110_2$$

↑ отбрасывается

Ответ: $1`\ 0010110_2 = -22_{10}$

б) $6_{10} + (-11)_{10} = -5_{10}$

$6_{10} \to 0`\ 0000110_2$
$-11_{10} \to 1`\ 0001011_2 \to 1`\ 1110100_{2\ обр} \to 1`\ 1110101_2$

$$\begin{array}{r} 0`\ 0000110_2 \\ +\ 1`\ 1110101_{2\ доп} \\ \hline 1`\ 1111011_{2\ доп} \end{array} \to 1`\ 0000100_{2\ обр} + 1`\ 0000101_2$$

Ответ: $1`\ 0000101_2 = -5_{10}$

в) $8 + (-3) = 5$

$8_{10} \to 0`\ 0001000_2$
$(-3_{10}) \to 1`\ 0000011_2 \to 1`\ 1111100_{2\ обр} \to 1`\ 1111101_{2\ доп}$

$$\begin{array}{r} 0`\ 0001000_2 \\ +\ 1`\ 1111101_{2\ доп} \\ \hline 10`\ 0000101_{2\ доп} \end{array}$$

↑ отбрасывается

Ответ: $1`\ 0000101_2 = 5_{10}$

Модифицированные обратный и дополнительный коды

При сложении может возникнуть ситуация, которая называется ***переполнением разрядной сетки***. Это происходит в том случае, когда сумма чисел больше либо равна 2^{n-1}, и тогда старший разряд суммы оказывается в знаковом разряде, т. е. происходит перенос единицы в знаковый разряд. Это приводит к неправильному результату, причем положительное число, получившееся в результате арифметической операции, может восприниматься как отрицательное (так как в знаковом разряде – 1) и наоборот.

Например, для однобайтового формата ($n = 8$, $2^{n-1} = 2^7 = 128$) при сложении чисел 67 и 89 их сумма больше 128, что приведет к переполнению разрядной сетки и неправильному результату.

$67_{10} \to 0`\ 1000011_2$
$89_{10} \to 0`\ 1011001_2$

$$\begin{array}{r} 0`\ 1000011_2 \\ +\ 0`\ 1011001_2 \\ \hline 1`\ 0011100_2 \end{array}$$

Для обнаружения переполнения и оповещения о возникновении ошибки вводятся модифицированные коды. В модифицированном обратном и модифицированном дополнительном кодах под знак числа отводится не один, а два разряда: «00» соответствует знаку «+», «11» — знаку «–». Любая другая комбинация («01» или «10»), получившаяся в знаковых разрядах, служит признаком переполнения разрядной сетки.

Сложение чисел в модифицированных кодах ничем не отличается от сложения в обычных обратном и дополнительном кодах.

Формы представления чисел в компьютере

Числа в компьютере хранятся в регистрах и ячейках памяти с ограниченным количеством разрядов. Для решения большинства задач числа внутри компьютера достаточно представить в виде целых и вещественных чисел. Представление целых чисел достаточно простое — выделяется необходимое количество разрядов для размещения чисел. Для представления вещественных чисел используются две формы записи: число с фиксированной точкой[1] (ЧФТ) и число с плавающей точкой (ЧПТ).

Форма записи числа с фиксированной точкой точно определяет положение точки между целой и дробной частью. Запись числа с фиксированной точкой обычно имеет знаковый и цифровой разряды. Фиксированная точка означает, что на этапе конструирования было определено, сколько и какие разряды машинного слова отведены под изображение целой и дробной частей числа. Точка в разрядной сетке может быть зафиксирована, в принципе, после любого разряда.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

знак числа — целая часть числа — положение точки — дробная часть числа

Рис. 2.40. Представление машинного слова с фиксированной точкой

Например, число $-12{,}25_{10} = -1100{,}01_2$ будет записано в виде машинного слова следующим образом, при этом в знаковом разряде ноль соответсвует знаку «+», а единица знаку «–».

1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0

знак числа — положение точки

Как частный случай числа с фиксированной точкой может быть рассмотрена запись целого числа (в этом случае все разряды, кроме знакового, используются для записи целой части). К достоинствам использования чисел с фиксированной точкой относятся простота выполнения арифметических операций и высокая точность изображения чисел. К недостаткам — небольшой диапазон представления чисел.

[1] Точка в американской системе соответствует запятой в русской.

Для представления чисел с плавающей точкой (ЧПТ) используется следующая форма записи числа:

$$N = \pm mq^{\pm p},$$

где q — основание системы счисления, p — порядок числа, m — мантисса числа N, представляющая собой правильную положительную дробь.

Положение точки определяется значением порядка p. С изменением порядка точка перемещается (плавает) влево или вправо. Так, например, число 156_{10} можно записать как

$156_{10} = 15.6 \cdot 10^1$
$156_{10} = 1.56 \cdot 10^2$
$156_{10} = 0.156 \cdot 10^3$
$156_{10} = 0.0156 \cdot 10^4$ и т. д.

Для установления однозначности при записи чисел введем ограничение — в первом разряде мантиссы стоит отличная от нуля цифра, т. е. мантисса может изменяться в диапазоне:

$$\frac{1}{q} \leqslant |m| \leqslant 1.$$

Такое представление числа называется **нормализованным**. Например, из всех возможных записей числа 156 (см. выше) нормализованная форма числа будет представлена как $0.156 \cdot 10^3$. Мантисса здесь, очевидно, равна 0.156, а порядок равен 3.

Рассмотрим примеры представления чисел в нормализованной форме.

Десятичная система:
$568 = 0.568 \cdot 10^3$ $(m = 0.568;\ p = 3)$
$0{,}00568 = 0.568 \cdot 10^{-2}$ $(m = 0.568;\ p = -2)$
$-1{,}785 = -0.1785 \cdot 10^1$ $(m = 0.1785;\ p = 1)$

Двоичная система:
$10111{,}01 = 0.1011101 \cdot 2^{101}$ $(m = 0.1011101;\ p = 101)$
$-111{,}011 = 0.111011 \cdot 2^{11}$ $(m = 0.111011;\ p = 11)$
$0{,}0001011 = 0.1011 \cdot 2^{-11}$ $(m = 0.1011;\ p = -11)$

Представление чисел в нормализованной форме для десятичной системы счисления не вызывает затруднений, что касается представления двоичных чисел, то необходимо помнить, что порядок числа — двоичное число. Например, число 10111.01_2 имеет порядок, равный 5 (10111) в десятичной системе, но в двоичной — $5_{10}=101_2$, следовательно, нормализованная форма — $0.1011101 \cdot 2^{101}$.

Для представления чисел в машинном слове выделяют группы разрядов для изображения мантиссы, порядка, знака порядка и знака числа (рис. 2.41). В этом случае машинное слово делится на два основных поля. В одном записывается мантисса числа, во втором — указывается порядок числа.

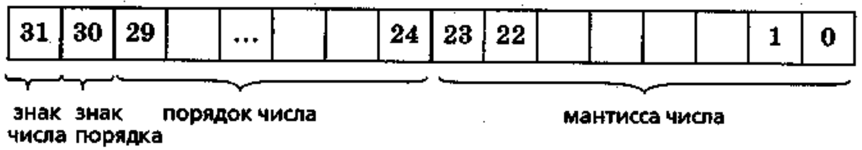
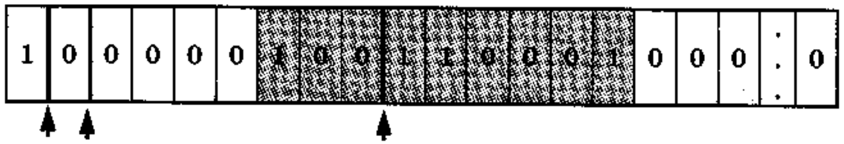


Рис. 2.41. *Представление 32-разрядного машинного слова с плавающей точкой*

Например, число $-12.25_{10} = -1100.01_2 = -0.110001 \cdot 2^{100}$ будет представлено следующим образом:



Таким образом, числа с плавающей точкой позволяют увеличить диапазон обрабатываемых чисел по сравнению с диапазоном чисел с фиксированной точкой. Однако быстродействие компьютера при обработке чисел с плавающей точкой гораздо ниже, чем при обработке чисел с фиксированной точкой.