

Арифметические основы цифровой техники

Лабораторная работа №8

Тематический план

1. Принципы построения системы счисления
2. Перевод чисел из одной системы в другую
3. Специальные системы счисления
4. Выбор системы счисления
5. Формы, диапазон и точность представления чисел
6. Представление информации в микропроцессорах класса Pentium
7. Кодирование отрицательных чисел
8. Арифметические операции сложения и вычитания
9. Арифметические операции умножения и деления

1 Принципы построения *системы счисления*

Числовая информация в компьютерах характеризуется:

- системой счисления (двоичная, десятичная и др.);
- видом числа (числа вещественные, комплексные, массивы);
- типом числа (смешанное, целое, дробное);
- формой представления числа (место запятой) — с природной (переменной), фиксированной, плавающей запятой;
- разрядной сеткой и форматом числа;
- диапазоном и точностью представления чисел;
- способом кодирования отрицательных чисел — прямым, обратным и дополнительным кодами;
- алгоритмами выполнения арифметических операций.



Системой счисления называется совокупность цифр и правил для записи чисел.

Запись числа в некоторой системе счисления называется его **кодом**.

Все системы счисления делятся на **позиционные и непозиционные**.

Для записи чисел в **позиционной** системе счисления используют определенное количество графических знаков (цифр и букв), которые отличаются один от другого. Число таких знаков q называют **основанием позиционной системы счисления**.



Таблица 1.1

<i>Основание q</i>	<i>Система счисления</i>	<i>Знаки</i>
2	Двоичная	0, 1
3	Троичная	0, 1, 2
5	Пятеричная	0, 1, 2, 3, 4
8	Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16	Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F



В **позиционных системах** счисления значение каждой цифры определяется ее изображением и позицией в числе.

Отдельные позиции в записи числа называют **разрядами**, а номер позиции - номером разряда.

Число разрядов в записи числа называется его **разрядностью** и совпадает с длиной числа.



В непозиционных системах счисления значение каждой цифры не зависит от ее позиции. Самой известной непозиционной системой является римская, в которой используются семь знаков — I, V, X, L, C, D, M, соответствующих таким значениям:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Например:

III — 3; LIX — 59; DLV — 555.

Недостатком непозиционной системы является отсутствие нуля и формальных правил записи чисел и соответственно арифметических действий с ними (хотя по традиции римскими числами часто пользуются при нумерации глав в книгах, ве в истории и др.).



Система счисления должна обеспечивать:

- возможность представления любого числа в заданном диапазоне;
- однозначность, сжатость записи числа и простоту выполнения арифметических операций;
- достижение высокого быстродействия в процессе обработки информации.



Число в позиционной системе можно представить полиномом

$$A_q = a_k \cdot q^k + a_{k-1} \cdot q^{k-1} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + \dots + a_{-m} \cdot q^{-m} = \sum_{i=-m}^k a_i \cdot q^i, \quad (1.6)$$

где q — основание системы счисления; q^i — вес позиции; $a_i \in \{0, 1, \dots, (q-1)\}$ — цифры в позициях числа; $0, 1, \dots, k$ — номера разрядов целой части числа; $-1, -2, \dots, -m$ — номера разрядов дробной части числа.

Позиционные системы с одинаковым основанием в каждом разряде называют однородными. Поскольку на значение q нет никаких ограничений, то теоретически возможно бесконечное множество позиционных систем счисления.



На практике применяют сокращенную запись полинома (1.6) в виде последовательности цифр со знаком в зависимости от типа числа:

- для смешанного числа

$$A_q = \pm a_k a_{k-1} \dots a_1 a_0, a_{-1} \dots a_{-m}; \quad (1.6, a)$$

- для целого числа

$$A_q = \pm a_k a_{k-1} \dots a_1 a_0; \quad (1.6, б)$$

- для правильной дроби

$$A_q = \pm 0, a_{-1} a_{-2} \dots a_{-m}; \quad (1.6, в)$$



Пример 1.1

Иллюстрация записи чисел в виде последовательности цифр (1.6) и соответствующего полинома (1.5):

- двоичная система: $q = 2$; $a_i \in \{0, 1\}$:

$$A_2 = 111,01 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 7,25_{10};$$

- восьмеричная система: $q = 8$; $a_i \in \{0, 1, 2, \dots, 7\}$:

$$A_8 = 45,21 = 4 \cdot 8^1 + 5 \cdot 8^0 + 2 \cdot 8^{-1} + 1 \cdot 8^{-2} = 37,265_{10};$$

- десятичная система: $q = 10$; $a_i \in \{0, 1, 2, \dots, 9\}$:

$$A_{10} = 135,64 = 1 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 4 \cdot 10^{-2}.$$

- шестнадцатеричная система: $q = 16$; $a_i \in \{0, 1, \dots, F\}$:

$$\begin{aligned} A_{16} = DE,1B &= D \cdot 16^1 + E \cdot 16^0 + 1 \cdot 16^{-1} + B \cdot 16^{-2} = \\ &= 13 \cdot 16^1 + 14 \cdot 16^0 + 1 \cdot 16^{-1} + 11 \cdot 16^{-2} = 222,105_{10}. \end{aligned}$$



Рассмотренные записи чисел показывают один из способов перевода недесятичных чисел в десятичные. При одинаковой разрядности в системах счисления с большим основанием можно записать больше разрядных чисел.

Достоинством двоичной системы является простота выполнения арифметических операций: наличие надёжных микроэлектронных схем с двумя устойчивыми состояниями (триггеров), предназначенных для хранения значений двоичного разряда – цифр 0 или 1. Двоичные цифры называют также битами. В двоично-десятичной системе счисления каждая десятичная цифра записывается четырьмя двоичными разрядами (тетрадами).



Пример 1.2

Запись десятичного числа в двоично-десятичной системе:

$$A_{10} = 873,25 = \overbrace{1000}^8 \overbrace{0111}^7 \overbrace{0011}^3, \overbrace{0010}^2 \overbrace{0101}^5.$$



2 Перевод чисел из одной системы в другую

Для перевода целого числа из одной системы счисления в другую необходимо разделить переводимое число на новое основание по правилам исходной системы. Полученный первый остаток является значением младшего разряда в новой системе, а первое частное необходимо снова разделить. Этот процесс продолжается вплоть до получения неделимого частного. Результат записывают в порядке, обратном их получению в виде формулы (1.6, б).



Пример 1.3

Перевод целого десятичного числа $A = 118$ в двоичное.

Исходное число

Частное

Остаток

118/2

59

0

59/2

29

1

29/2

14

1

14/2

7

0

7/2

3

1

3/2

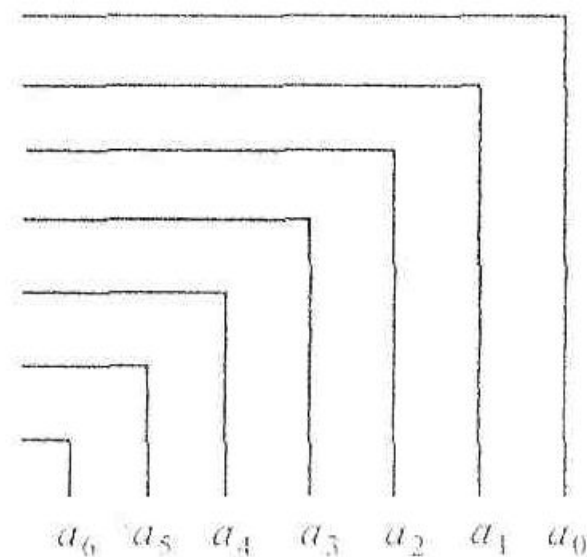
1

1

1/2

—

1



Результат: $A = 118_{10} = 1110110_2$.



Для перевода правильной дроби из одной системы счисления в другую необходимо, действуя по правилам исходной системы, умножить переводимое число на основание новой системы; от результата отделить целую часть, а оставшуюся дробную часть снова умножить на это основание. Процесс такого умножения повторяется до получения заданного числа цифр. Результат записывают как целые части произведения в порядке их получения, располагаемых в виде формулы (1.6, е).



Пример 1.4

Перевод правильной десятичной дроби $A = 0,625$ в двоичное число с точностью до четвертого знака:

0,	625	
	2	0;
1	250	
	2	$a_{-1} = 1;$
0	500	
	2	$a_{-2} = 0;$
1	000	
	2	$a_{-3} = 1;$
0	000	$a_{-4} = 0.$

Результат: $A = 0,625_{10} = 0,1010_2.$

Для перевода смешанных чисел в двоичную систему требуется отдельно переводить их целую и дробную части. В записи результата целая часть перевода отделяется от дробной в соответствии с формулой (1.6, а).



Пример 1.5

Перевод смешанного десятичного числа $A = 118,625_{10}$ в двоичное. Используя результаты перевода целого числа 118 в примере 1.3 и дробного числа в примере 1.4, запишем результат:

$$A = 118,625_{10} = 1110110,1010_2.$$

В восьмеричных и шестнадцатеричных числах основание системы счисления кратно степени двойки: $2^3 = 8$; $2^4 = 16$. По этой причине перевод этих чисел в двоичные реализуется очень просто: каждую цифру записывают тремя двоичными цифрами (триадами) для восьмеричных чисел и четырьмя двоичными цифрами (тетрадами) для шестнадцатеричного числа в направлении влево и вправо от запятой. При этом крайние незначащие нули опускаются.



Пример 1.6

Иллюстрация перевода восьмеричных и шестнадцатеричных чисел в двоичные:

$$A = 305,42_8 = \overbrace{011}^3 \overbrace{000}^0 \overbrace{101}^5, \overbrace{100}^4 \overbrace{010}^2_2;$$

$$A = 7AB,EF_{16} = \overbrace{0111}^7 \overbrace{1010}^A \overbrace{1011}^B, \overbrace{1110}^E \overbrace{1111}^F_2.$$

Для перевода двоичного числа в восьмеричное исходное число разбивают на триады влево от запятой, отсутствующие крайние цифры дополняют нулями. Затем каждую триаду записывают восьмеричной цифрой. Аналогично осуществляется перевод двоичного числа в шестнадцатеричное, при этом выделяют тетрады, заменяемые шестнадцатеричными цифрами.



Пример 1.7

Иллюстрация перевода двоичных чисел в восьмеричные и шестнадцатеричные:

$$A = 110011,100010_2 = \overbrace{110}^6 \overbrace{011}^3, \overbrace{100}^4 \overbrace{010}^2_2 = 63,42_8;$$

$$A = 110011,100010_2 = \overbrace{001}^3 \overbrace{100}^3 \overbrace{11,1000}^8 \overbrace{1000}^8_2 = 33,88_{16}.$$

Рассмотрим правила, облегчающие расчеты с двоичными числами.

1. Необходимо знать представление десятичных чисел от нуля до 15 в системах счисления с основанием $q = 2, 8, 16$ и 2-10 (табл. 1.2).
2. Необходимо знать десятичные значения чисел 2 от $k = 0$ до $k = 10$ (табл. 1.3).
3. Полезно знать следующие соотношения для двоичных целых и дробных чисел:



Таблица 1.2

$q = 10$	$q = 2$	$q = 8$	$q = 16$	$q = 2-10$
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0101
15	1111	17	F	0001 0101



Таблица 1.3

k	0	1	2	3	4	5	6	7	8	9	10
2^k	1	2	4	8	16	32	64	128	256	512	1024

$$A = \underbrace{100 \dots 00}_k = 2^k, \text{ например, } A = 1000_2 = 2^3 = 8_{10};$$

$$A = \underbrace{11 \dots 11}_k = 2^k - 1, \text{ например, } A = 11111_2 = 2^5 - 1 = 31_{10};$$

$$A = 0, \underbrace{11 \dots 11}_k = 1 - 2^{-k}, \text{ например, } A = 0,1111_2 = 1 - 2^{-4} = \left(\frac{15}{16}\right)_{10};$$

$$A = 0, \underbrace{00 \dots 01}_k_{(2)} = 2^{-k}_{(10)}, \text{ например, } A = 0,00001_{(2)} = 2^{-5}_{(10)} = \left(\frac{1}{32}\right)_{10}.$$



Двоичные дробные числа в десятичном виде записываются так: k разрядов справа от запятой определяют целое положительное число в числителе, а в знаменателе — число 2^k . Например, $A = 0,1011_2 = (11/2^4)_{10} = (11/16)_{10}$, что читается как "одиннадцать шестнадцатых".



3 Специальные системы счисления

Рассмотренные позиционные системы счисления относят к классическим. Кроме них, в компьютерах используют ряд специальных позиционных двоичных систем, в том числе:

- систему с использованием символов 0, -1, или 1,-1; для удобства символ -1 обозначают как $\bar{1}$; систему с символами 1 и $\bar{1}$ называют системой счисления $(1, \bar{1})$;

- системы с отрицательным основанием $q < -1$ и символами 0, 1, ..., $(q-1)$;

систему с основанием $q = -2$ и символами 0, 1 называют **минус-двоичной**;

- избыточную систему с основанием q и количеством символов больше q , например, при $q = 2$ используют символы $(1, 0, \bar{1})$, такую систему называют также **симметричной знакоразрядной**.



Преимуществом специальных систем счисления является упрощение и ускорение выполнения ряда арифметических операций (сложение, вычитание, умножение, деление) и представление единым кодом положительных и отрицательных чисел без дополнительного знакового разряда. Недостатком специальных систем является сложность правил перевода их в классические системы счисления и наоборот, а также неоднозначное представление ряда чисел.

Если в формуле (1.6) учесть, что символы a_i , принимают значение 1 или $\bar{1}$, то для основания $q=2$ число $A=99_{10}$, будет иметь вид $111\bar{1}\bar{1}\bar{1}1$. В системе чисел $(1, \bar{1})$ отсутствует символ "нуль", поэтому некоторые целые и дробные числа в ней не могут быть представлены конечным множеством символов. Например, десятичное число $A=20_{10}$ в системе чисел $(1, \bar{1})$ записывается как бесконечное число $11\bar{1}1\bar{1},\bar{1}\bar{1}\bar{1}...$, что обуславливает погрешность в системе $(1, \bar{1})$. В то же время есть числа, которые не имеют единого изображения. 

В минус-двоичной системе может быть представлено любое положительное или отрицательное число. Например, для, $q=-2$, $n=4$ с учетом формулы (1.6):

$$A=5_{10}=0101_{(-2)}; A=5_{10}=-5=1111_{(-2)}.$$

Некоторые числа представлены неоднозначно и только бесконечными дробями: $A = 1/3 = 0,010101\dots$ или $A = 1,101010\dots$

Избыточная двоичная система связана с обычной соотношением:

$$\underbrace{11\dots11}_k_{(2)} = \sum_{i=0}^{k-1} 2^i_{(10)} = 2^k - 1 = 1\underbrace{00\dots01}_k.$$



На основании выражения (1.6) осуществляется переход от обычной двоичной системы к избыточной с символами 1,0, $\bar{1}$ и наоборот. Избыточная система имеет следующие свойства:

- 1) одно и то же число не имеет единой записи: $A = 12_{10} = 01100_2 = 10\bar{1}00_2 = 1\bar{1}100_2$;
- 2) положительные и отрицательные числа изображаются без использования дополнительного знакового разряда: $A = 13_{10} = 01101_2 = 10\bar{1}01$; $A = -13_{10} = -01101_2 = \bar{1}010\bar{1}$ (при смене знака достаточно заменить в записи числа 1 на $\bar{1}$ и наоборот);
- 3) возможно уменьшение количества единиц в записи числа, что позволяет упростить и ускорить выполнение арифметических операций, особенно сложение и вычитание.



Преобразование двоичных чисел в избыточную систему с минимальным числом единиц в разрядах осуществляют так. На первом этапе все комбинации в записи числа вида 01...11 заменяют равнозначной комбинацией вида 10...01 с тем же числом разрядов; затем выполняют замену сочетаний $1\bar{1}$ на 01 и $\bar{1}1$ на $0\bar{1}$, например:

$$\begin{aligned}
 A &= 10\underbrace{01}_1\underbrace{11}_2\underbrace{00}_3\underbrace{11}_2\underbrace{01}_3\underbrace{10}_2\underbrace{10}_2\underbrace{10}_2\underbrace{10}_2 \quad (2) = 10\underbrace{100\bar{1}}_1\underbrace{0100\bar{1}}_2\underbrace{110\bar{1}}_3\underbrace{01}_2 = \\
 &= 10100\bar{1}0100\bar{1}10\bar{1}01 = 10100\bar{1}01\underbrace{000\bar{1}}_{1^*}\underbrace{0101}_{1^*}.
 \end{aligned}$$

Количество единиц в записи числа уменьшилось с десяти до четырех.

Она применяется в троичной логике и была технически реализована в вычислительной машине «Сетунь».



4 Выбор системы счисления

В компьютерах в основном используют однородные позиционные системы счисления. При выборе основы этих систем учитывают такие показатели:

1. Наличие физических элементов для изображения цифр системы в виде одного из q состояний, например, разницы напряжений. Уменьшение числа состояний упрощает физический элемент, поэтому наиболее приемлемой является двоичная система.
2. Экономичность системы счисления. Система с большим основанием q обеспечивает представление определенного числа меньшим количеством разрядов. Однако при этом усложняется построение физического элемента с большим числом состояний.



Эффективность системы счисления с основанием q , оценивается количеством цифровых разрядов D , необходимых для изображения определенного числа с длиной n . При этом учитывается, что уменьшение значения D приводит к уменьшению электронных схем для представления чисел в компьютерах.

Для представления любого десятичного числа длиной, например, $n_{10}=6$ используют $D_{10}=q_{10}*n_{10}=10*6=60$ цифроразрядов. В двоичной системе для представления того же числа (с учетом соотношения $n_2=3*n_{10}$) требуется иметь $D_2=q_2*3*n_{10}=2*3*6=36$ цифровых разрядов.



Наиболее экономичной является система с основанием $q=2,73... \approx 3$. Двоичная система по экономичности уступает троичной на 5,8%, однако имеет более надежные физические элементы. Кроме того, для запоминания цифр троичной системы 0. 1.2 используют два двоичных физических элемента. Из этого следует, что самой эффективной является двоичная система счисления.

3. Трудоемкость и быстродействие выполнения арифметических операций. Чем меньше основание q , тем меньше цифр принимают участие в вычислении данных и тем выше быстродействие компьютера. Например, быстродействие машины в двоичной системе превышает быстродействие в троичной на 26,2%, а в десятичной – в 2.7 раза.

4. Наличие формального математического аппарата для анализа и синтеза цифровых схем. Таким аппаратом для двоичных элементов является булева алгебра.



Таким образом, из перечисленных показателей видно, что самой приемлемой для применения в компьютерах является однородная позиционная двоичная система счисления. Двоичные системы счисления используют в больших и средних компьютерах, предназначенных для решения научно-технических задач с большим объемом вычислений и сравнительно малым количеством исходных данных.



Двоично-десятичную систему применяют для решения экономических задач, которые характеризуются большим объемом входных и выходных данных в сравнении с малым объемом вычислений.

Двоично-десятичная система имеет следующие достоинства:

- * не требуется перевод исходных данных из одной системы в другую;
- * удобство контроля результатов изображением их на экране дисплея;
- * удобство автоматического контроля через наличие избыточных кодов в изображении цифр: 1010, 1011, . . , 1111.



5 Формы, диапазон и точность представления чисел

Двоичные числа в компьютерах размещаются в ячейках памяти или в регистрах, которые состоят из запоминающих элементов - триггеров. В ячейке или триггере хранится значение одного двоичного разряда - бит информации. **Разрядной сеткой компьютера** называется совокупность запоминающих элементов для размещения одного двоичного числа. Для разных классов компьютеров длина разрядной сетки составляет 8, 16, 32, 64 и больше разрядов. **Форматом** называется способ размещения компонентов числа в разрядной сетке, то есть последовательность и позиции знака, мантиссы, порядка и др.



В компьютерах используются две формы представления числа:

- с фиксированной запятой перед старшим разрядом числа (для правильной дроби) или после младшего (для целого числа);
- с плавающей запятой, место положения которой задается порядком числа. Место запятой в обоих форматах понимается неявно, без использования дополнительных разрядов. По традиции нумерация разрядов в больших машинах осуществляется слева направо, а в мини- и микрокомпьютерах и микропроцессорах - справа налево.

В числах со знаком выделяют дополнительный знаковый разряд (обычно, крайний слева). В нем для знака плюс записывают цифру 0, а для знака минус - цифру 1. В числах без знака все разряды числа определяют модуль. Форматы двоичных дробных и целых чисел со знаком, с нумерацией разрядов справа налево и значением их веса показаны на рис. 1.6.



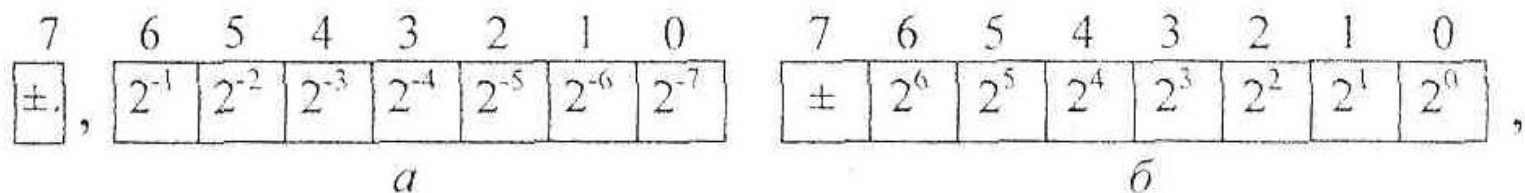


Рис. 1.6. Формат байта: а — для правильной дроби; б — для целого числа

Числа в разрядную сетку записывают в соответствии с весом позиций. Не занятые разряды заполняют нулями, как это показано на рис. 1.7 для чисел $+0,725_{10} \approx +0,10111_2$ и $-47_{10} = -101111_2$.

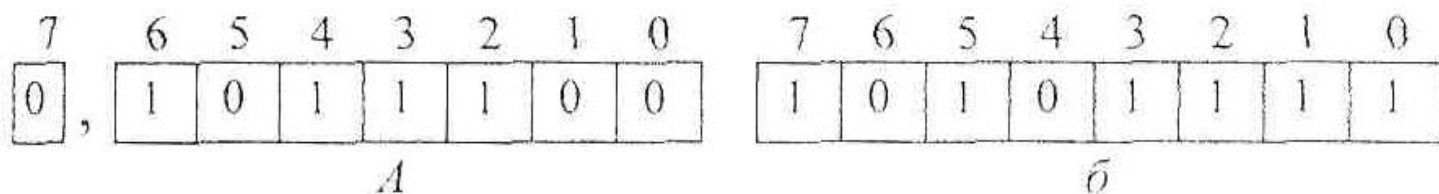


Рис. 1.7. Запись чисел: а — дроби $+0,10111_2$; б — целого числа -101111_2



Интервал числовой оси, заключенный симметрично между максимальным и минимальным значением числа, называется **диапазоном представления**. Абсолютное значение целого числа A изменяется в пределах:

$$A_{\min} \leq |A| \leq A_{\max}; \quad A_{\min} = 1; \quad A_{\max} = 2^k - 1, \quad (1.7)$$

где k — количество разрядов цифровой части числа.

Из выражения (1.7) следует, что числа, по абсолютному значению меньше единицы, воспринимаются как "машинный нуль"; числа, больше $A_{\max}$, вызывают переполнение разрядной сетки. Диапазон представления целых чисел с учетом симметрии относительно нуля числовой оси

$$D_A = 2A_{\max} = 2(2^k - 1) \approx 2^{k+1}. \quad (1.8)$$

Пример 1.8

Рассчитать максимальное значение и диапазон представления целого числа со знаком в 16-разрядной сетке. На основе выражений (1.7) и (1.8) для $k = 15$ получаем:

$$A_{\max} = 2^{15} - 1 = 32767; \quad D = 2A_{\max} = 65534.$$



Абсолютное значение правильной дроби B изменяется в пределах:

$$B_{\min} \leq |B| \leq B_{\max}; \quad B_{\min} = 2^{-k}; \quad B_{\max} = 1 - 2^{-k}. \quad (1.9)$$

Правильная дробь, по модулю меньше 2^{-k} , воспринимается как "машинный нуль"; числа, больше единицы, вызывают переполнение разрядной сетки. Диапазон представления правильной дроби

$$D_B = 2B_{\max} = 2(1 - 2^{-k}) \approx 2,$$

так как $1 \gg 2^{-k}$.

Точность представления чисел в компьютере определяется абсолютной и относительной погрешностью. Абсолютная погрешность представления чисел в компьютере составляет:

для целых чисел

$$\Delta A = 1/2 = 2^{-1}; \quad (1.10)$$

для правильной дроби

$$\Delta B = 2^{-k}/2 = 2^{-(k+1)}. \quad (1.11)$$

Из выражения (1.10) следует, что абсолютная погрешность представления целых чисел в форме с фиксированной запятой равна половине младшего разряда и не зависит от длины разрядной сетки. Из выражения (1.11) видно, что абсолютная погрешность представления правильной дроби в форме с фиксированной запятой уменьшается при увеличении длины разрядной сетки.



Относительная погрешность δ представления числа задается отношением абсолютной погрешности к текущему значению числа и находится в пределах

$$\delta_{\min} \leq \delta \leq \delta_{\max}.$$

Для целых чисел в форме с фиксированной запятой минимальное и максимальное значения относительной погрешности

$$\begin{aligned}\delta_{A\min} &= \Delta A / A_{\max} = 2^{-1} / (2^k - 1) \approx 2^{-(k+1)}; \\ \delta_{A\max} &= \Delta A / A_{\min} = 2^{-1} / 1 = 0,5.\end{aligned}$$



Таковыми же формулами определяются минимальное и максимальное значения относительной погрешности для правильной дроби с фиксированной запятой. Относительная погрешность представления чисел в форме с фиксированной запятой существенно зависит от значения числа и значительно возрастает для малых чисел. В форме с плавающей запятой числа представляются в виде произведений:

$$C = q^P \cdot M = X \cdot M \quad (1.12)$$

где (q — основание системы счисления (обычно целая степень числа 2); P — порядок числа длиной $k+1$ (целое число со знаком); M - мантисса числа длиной $r+1$ (правильная дробь со знаком); X - характеристика числа.



Знак всего числа определяется знаком мантиссы. Для мини-компьютеров основания порядка и мантиссы совпадают (далее предполагается этот случай), а для больших машин – они различны. Представление числа формулой (1.12) называют также полулогарифмическим, потому что только часть числа – характеристика – представляется в логарифмической форме.

Мантисса называется нормализованной, если ее значение определяется неравенством вида

$$1/q \leq |M| < 1 \quad (1.13)$$

то есть в старшем разряде модуля мантиссы должна быть записана единица. Так, для основания $q=2$ имеем $2^{-1} \leq |M| < 1$.

Пример 1.9

Иллюстрация записи числа в форме с плавающей запятой:

$$A_2 = 2^1 \cdot 110,111 = 2^2 \cdot 11,0111 = 2^4 \cdot 0,110111.$$



Значение порядка указывает на количество позиций, на которые "плавает" запятая.

Формат числа с плавающей запятой в 16-разрядной сетке показан на рис. 1.8. Тут для модулей порядка и мантиссы отведено соответственно пять и девять разрядов. Запятая в порядке размещена (условно) после младшего разряда, а в мантиссе - перед старшим. Знаки порядка и мантиссы размещены перед их старшими разрядами.

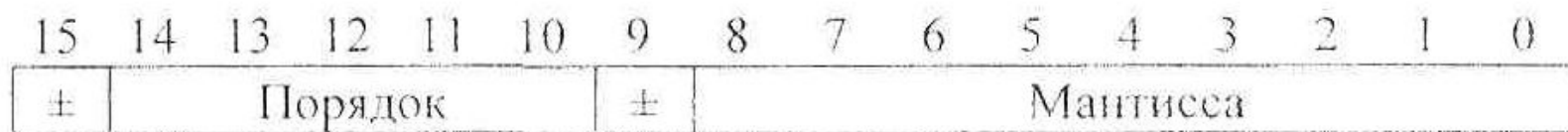


Рис. 1.8. Формат чисел в форме с плавающей запятой



Абсолютное значение числа C' в форме с плавающей запятой с учетом формул 1.7), (1.9) и (1.13) изменяется в пределах

$$X_{\min} M_{\min} = C_{\min} \leq |C'| \leq C_{\max} = X_{\max} M_{\max}, \quad (1.14)$$

где $X_{\min} = 2^{-\Pi_{\max}}$; $M_{\min} = 2^{-1}$; $X_{\max} = 2^{+\Pi_{\max}}$; $M_{\max} = 1 - 2^{-k}$; $\Pi_{\max} = 2^r - 1$; и k — разрядность модулей порядка и мантиисы.

Диапазон представления чисел в форме с плавающей запятой

$$D_c = 2C_{\max} = 2 \cdot 2^{+\Pi_{\max}} (1 - 2^{-k}) \approx 2^{+\Pi_{\max} + 1}. \quad (1.15)$$

Пример 1.10

Рассчитать значение диапазона D_c для числа C' при $r = 5$, $k = 9$. С учетом выражения (1.15) получаем:

$$\Pi_{\max} = 2^r - 1 = 2^5 - 1 = 31; \quad D_c = 2^{31+1} = 2^{32},$$

что приблизительно соответствует десятичному числу $10^{32 \cdot 0.3} \approx 10^9$. Диапазон представления чисел с плавающей запятой приблизительно больше в $\Pi_{\max}$ раз диапазона представления чисел в форме с фиксированной запятой.



Абсолютная погрешность представления чисел в форме с плавающей запятой зависит от погрешности мантиссы и порядка числа:

$$\Delta C = \Delta M \cdot 2^{\pm n}; \quad \Delta M = 2^{-(k+1)}, \quad (1.16)$$

где ΔM — погрешность представления мантиссы.

Минимальная и максимальная относительные погрешности представления чисел в форме с плавающей запятой не зависят от характеристики (она записывается в числителе и знаменателе выражения и потому сокращается). С учетом формул (1.14) и (1.16) относительные погрешности рассчитывают из соотношений:

$$\begin{aligned} \delta C_{\min} &= \frac{\Delta M}{M_{\max}} = \frac{2^{-(k+1)}}{1 - 2^{-k}} \approx 2^{-k+1}; \\ \delta C_{\max} &= \frac{\Delta M}{M_{\min}} = \frac{2^{-(k+1)}}{2^{-1}} = 2^{-k}. \end{aligned} \quad (1.17)$$

Из выражений (1.17) следует, что относительные погрешности представления чисел в форме с плавающей запятой практически постоянны во всем диапазоне чисел.



6 Представление информации в микропроцессорах класса Pentium

Рассмотрим представления операндов в 32-разрядных микропроцессорах класса Pentium. В них используются такие типы данных: целые числа, вещественные числа, двоично-десятичные числа и строки битов, байтов и слов. Целые числа представляются со знаком и без знака в форматах байта, слова, двойного и учетверенного слова длиной соответственно 8, 16, 32 и 64 бит (рис. 1.9).



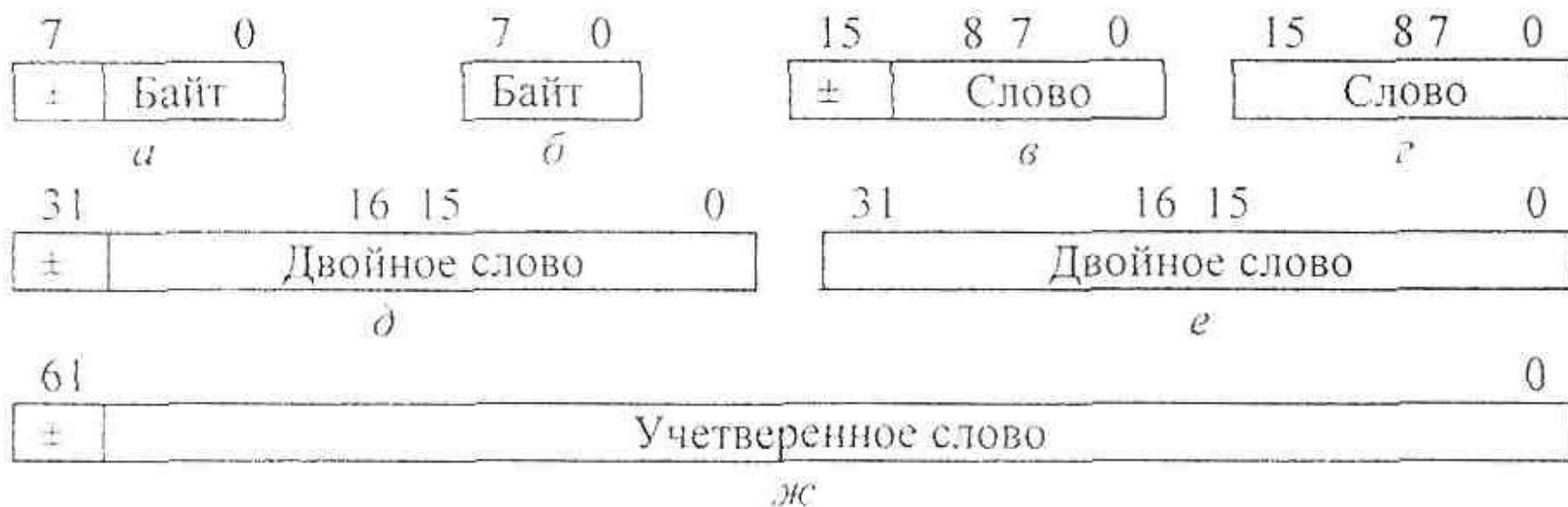


Рис. 1.9. Форматы целых чисел: а, в, д, ж — со знаком; б, г, е — без знака



Над числами в этих форматах выполняются операции сложения, вычитания, умножения, деления и др. Диапазоны представления знаковых и беззнаковых значений в данных форматах представлены в табл. 1.4

Таблица 1.4.

Формат числа	Диапазон представления чисел	
	без знака	со знаком
Байт	0...255	-128...+127
Слово	0...65535	-32768 ...+32767
Двойное слово	$0...4*10^9$	$-2*10^9...+2*10^9$
Учетверенное слово	$0...5*10^{19}$	$-2,5*10^{19} ...+2,5*10^{19}$



Вещественные числа представлены в формате с плавающей запятой в коротком (32 бит), длинном (64 бит) и расширенном (80 бит) форматах (рис. 1.10). Числа с плавающей запятой длиной 32 и 64 бит, которые используются во многих компьютерах, например, в ЕС ЭВМ, СМ ЭВМ, обычно называют числами с одинарной и двойной точностью. Расширенный формат характерен только для процессоров класса Pentium.

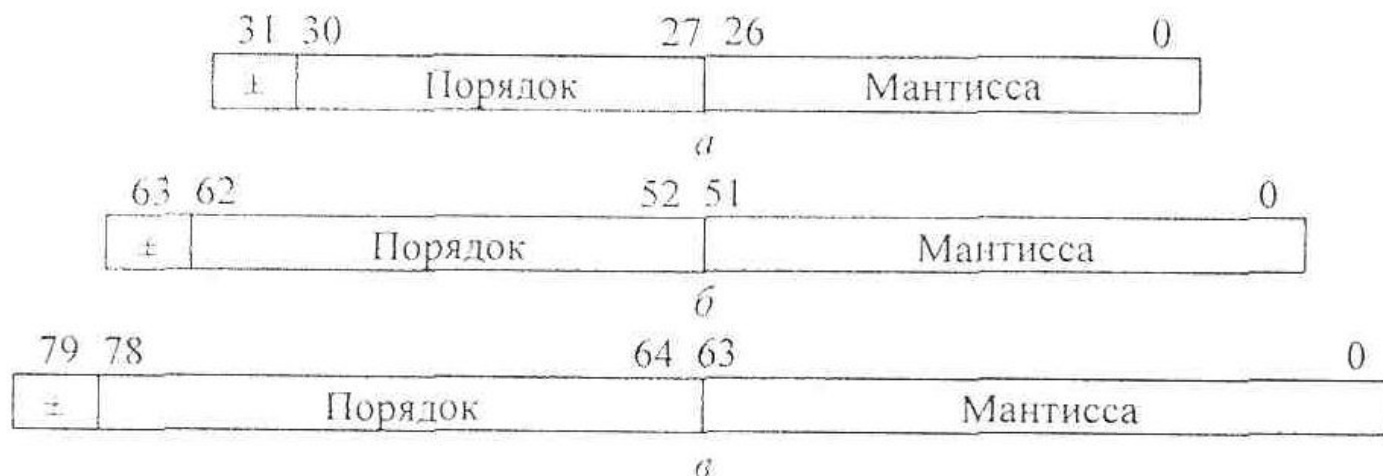


Рис. 1.10. Форматы чисел с плавающей запятой; *a* — короткий; *b* — длинный; *v* — расширенный



В данных форматах чисел с плавающей запятой используется смещенные порядки $P_{см}$:

$$P_{см} = P + \Delta P = P + 2^{r-1} - 1 = \begin{cases} P + 127 & \text{для } r = 8; \\ P + 1023 & \text{для } r = 11; \\ P + 16383 & \text{для } r = 15. \end{cases}$$

где P – значение истинного порядка; $\Delta P = 2^{r-1} - 1$ – смещение; r – длина порядка, которая равна 8, 11 или 15 бит соответственно для короткого, длинного и расширенного формата. Значение $P_{см}$ всегда положительное, поэтому знаковый разряд не нужен. Представление порядка со смещением упрощает операции сравнения чисел с плавающей запятой, что особенно важно для алгоритмов сортировки.



Значение числа с плавающей запятой и смещенным порядком определяется по формуле

$$C = (-1)^S 2^{P_{\text{см}} - \Delta P} (F_0, F_1, \dots, F_i, \dots, F_n),$$

где S — знак числа; n — число, которое для разных форматов равно 23, 52 или 64.

В машине мантисса представлена в нормализованной форме, которая состоит из целой части $F_0 = 1$ и дроби в таком виде:

$$M = 1, F_1, F_2, \dots, F_i, \dots, F_n.$$

В коротком и длинном форматах бит F_0 , при передаче чисел и хранении их в памяти не фигурирует. Это — скрытый (неявный) бит, который в нормализованном числе всегда равен единице.



Пример 1.11

Представить десятичное число -247,375 в коротком формате. Двоичный код этого числа равен -11110111,011; $-11110111,011 = -1,1110111011 \cdot 2^7$ истинный порядок будет +7 (запятая сдвигается влево на семь разрядов), а смещение достигнет значения

$P_{\text{см}} = 127 + 7 = 134$. С учетом скрытого бита $F_0 = 1$ имеем:

Знак	Порядок	Мантисса
1	10000110	1110 1110 1100 0000 0000 0000 0000



Параметры форматов вещественных чисел представлены в табл. 1.5.

Таблица 1.5

Параметры	Формат		
	короткий	длинный	расширенный
Длина формата, бит	32	64	80
Длина мантиссы, бит	24	53	65
Длина порядка, бит	8	11	15
Смещение порядка	+127	+1023	+16383
Диапазон	10^{+38}	10^{+308}	10^{+4932}



Числа в коротком и длинном форматах существуют только в памяти. При загрузке чисел в одном из этих форматов в микропроцессор они автоматически преобразуются в 80-битный формат, который используется только для внутренних операций. Аналогично данные из процессора преобразуются в короткий или длинный формат для хранения в памяти.

Точность вычислений чисел с плавающей запятой возрастает с увеличением длины мантиссы. Диапазон представления чисел с плавающей запятой зависит от длины порядка и основания счисления q . В машинах ЕС ЭВМ значение $q \sim 16$. В процессорах Pentium диапазон представления чисел в коротком формате для $q=2$ находится в пределах $10^{\pm 38}$. Если же для этого формата взять основание $q=16$, то получим значение диапазона в пределах $10^{\pm 152}$.



В микропроцессорах Pentium используются двоично-десятичные цифры в таких форматах:

- восьмиразрядные упакованные, которые содержат в одном байте две десятичные цифры в коде 8421, например, $39_{(10)} = 0011\ 1001_{(2-10)}$;
- восьмиразрядные неупакованные, которые содержат одну десятичную цифру в байте (младшая тетрада) вместе с признаком (зоной) 00112 в международном коде ASCII, например, $39_{(10)} = 0011\ 0011\ 0011\ 1001_{(2-10)}$. Представление десятичного числа 136492 в неупакованном и упакованном форматах показано на рис. 1.11;
- 80-разрядные упакованные, в которые записываются 19 десятичных цифр и в старшую тетраду записывается знак числа.



7 0 7 0 7 0 7 0 7 0 7 0

0011 0001	0011 0011	0011 0110	0011 0100	0011 1001	0011 0010
-----------	-----------	-----------	-----------	-----------	-----------

а

7 0 7 0 7 0

0001 0011	0110 0100	1001 0010
-----------	-----------	-----------

б

Рис. 1.11. Представление десятичных цифр в форматах:
а - неупакованном; б – упакованном



7 Кодирование отрицательных чисел

Для записи знака числа, замены операции вычитания чисел сложением их кодов, а также для определения переполнения разрядной сетки используют прямой обратный и дополнительный коды, где для представления знака числа отводится знаковый разряд, который располагается слева от числа и отделяется запятой. В знаковый разряд записывают нуль (для положительного числа) или единицу (для отрицательного). Запятая в машине в явном виде не изображается, а только подразумевается.



Числа, представленные в прямом, обратном и дополнительном кодах, называют машинными изображениями. Они состоят из знакового разряда и цифровой части (модуля числа). Положительные числа во всех кодах записываются одинаково. Если обозначить машинные изображения числа A в прямом коде $[A]_{\text{пр}}$, в обратном коде $[A]_{\text{об}}$, в дополнительном коде $[A]_{\text{д}}$, то для положительных чисел имеем:

$$[A]_{\text{пр}} = [A]_{\text{об}} = [A]_{\text{д}}.$$



Для отрицательных чисел с длиной мантииссы k битов коды определяют по таким правилам:

прямой код:

$$[A]_{np} = \begin{cases} 1 + |A| - \text{для дроби}; \\ 2^k + |A| - \text{для целого числа}; \end{cases}$$

обратный код:

$$[A]_{об} = \begin{cases} 2 - |A| - 2^{-k} - \text{для дроби}; \\ 2^{k-1} - |A| - \text{для целого числа}; \end{cases}$$

дополнительный код:

$$[A]_{д} = \begin{cases} 2 - |A| - \text{для дроби}; \\ 2^{k+1} - |A| + 1 - \text{для целого числа}; \end{cases}$$



Пример 1.12

Запись двоичных чисел $A = 1010_2$ и $B = -1010_2$ в прямых кодах: $[A]_{\text{пр}} = 0,1010$; $[B]_{\text{пр}} = 1,1010$.

Обратный код отрицательного числа образуется из его прямого кода после инвертирования значений разрядов цифровой части, то есть заменой нуля на единицу и единицы – на нуль; значение знакового разряда не изменяется. Дополнительный код отрицательного двоичного числа образуется из его обратного кода путем добавления единицы к младшему разряду.

Пример 1.13

Представление в обратном и дополнительном кодах двоичных чисел: целого $A = -1001_2$ и дробного $B = -1001_2$:

$$[A]_{\text{об}} = 1,0110; [A]_{\text{д}} = 1,0111; [B]_{\text{об}} = 1,0110; [B]_{\text{д}} = 1,0111.$$



Из этих примеров видно, что обратный и дополнительный коды целых и дробных чисел по виду записи совпадают; расхождения между ними отражены в алгоритмах обработки информации.

В модифицированных кодах знак числа дублируется в двух знаковых разрядах.

Пример 1.14

Представление в модифицированных кодах чисел $A = 1101_2$ и $B = -1101_2$: $[A]_{\text{ПР}} = [A]_{\text{ОБ}} = [A]_{\text{Д}} = 00,1101$;
 $[B]_{\text{ПР}} = 11,1101$; $[B]_{\text{ОБ}} = 11,0010$; $[B]_{\text{Д}} = 11,0011$.

Для перехода от обратного кода отрицательного числа к прямому коду требуется инвертировать значения разрядов цифровой части, не изменяя значения знакового разряда. Для перехода от дополнительного кода отрицательного числа к прямому сначала получают его обратный код, а затем добавляют единицу к младшему разряду.



8 Арифметические операции

сложения и вычитания

Все операции в компьютере выполняются в арифметико-логическом устройстве (АЛУ). Числа, которые участвуют в операциях, называются *операндами*. Основной операцией в АЛУ является сложение. Операция вычитания заменяется сложением операндов в обратном или дополнительном кодах. Операции умножения и деления сводятся к многократным сложениям и сдвигам.

Правила выполнения операций сложения, вычитания, умножения и деления по модулю 2 в двоичной арифметике представлены в табл. 1.6. При сложении двух единиц возникает перенос в старший разряд; при вычитании из нуля единицы требуется заем из старшего разряда.



Таблица 1.6

Сложение	Вычитание	Умножение	Модуль 2
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$	$0 + 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 * 1 = 0$	$0 + 1 = 1$
$1 + 0 = 1$	$1 - 1 = 0$	$1 * 0 = 0$	$1 + 0 = 0$
$1 + 1 = \underline{1}0$ Перенос	$0 - 1 = \underline{1}1$ Заем	$1 * 1 = 1$	$1 + 1 = 0$



Операция сложения в АЛУ выполняется типовым функциональным узлом — *сумматором*. Он состоит из отдельных логических схем, называемых одноразрядными сумматорами. Связь между отдельными одноразрядными сумматорами реализуется с помощью цепей переноса в направлении от младших разрядов к старшим.

Операнды А и В поступают с выходов регистров операндов. Комбинационный сумматор строится на логических элементах и не обладает свойством запоминания информации, поэтому результат сложения записывается в регистр результата.



Во всех компьютерах имеются команды сложения и вычитания чисел. Однако в сумматорах реализуются только операции сложения условно положительных машинных изображений. Машинные изображения положительных операндов во всех кодах совпадают. Машинные изображения отрицательных операндов получают по правилам представления чисел в обратном и дополнительном кодах. В операциях вычитания знак второго операнда (вычитаемого) автоматически изменяется на противоположный и после этого получают его машинное изображение. Поэтому в последующих примерах рассматриваются, только операции сложения.



В машинных изображениях (для обратного и дополнительного кодов) знаковый разряд и цифровая часть числа рассматриваются как одно целое. Они одинаково участвуют в операции сложения. При сложении в обратных кодах перенос из старшего знакового разряда результата поступает на вход переноса младшего разряда (циклический перенос). При сложении в дополнительных кодах перенос из старшего знакового разряда результата не учитывается, поэтому в сумматоре цепь циклического переноса разрывается. Знак результата при сложении машинных изображений образуется автоматически.



При сложении двоичных n -разрядных чисел $A = a_n, \dots, a_i, \dots, a_1$ и $B = b_n, \dots, b_i, \dots, b_1$ результат в каждом разряде определяется по формулам:

$$a_i + b_i + Z_i = S_i + 2 \cdot P_{i+1},$$

$$P_{i+1} = \begin{cases} 0, & \text{при } a_i + b_i \leq 1; \\ 1, & \text{при } a_i + b_i \geq 2; \end{cases}$$

где a_i, b_i - значения i -х разрядов; Z_i - перенос из младшего разряда; S_i - результат; P_{i+1} - перенос в старший разряд.



Пример 1.15

Сложить двоичные числа $A = 1010$ и $B = 0011$ в обратном коде:

$$\begin{array}{r} [A]_{OB} = 0,1010 \\ + [B]_{OB} = 0,0011 \\ \hline \end{array}$$

$$[C]_{OB} = 0,1101;$$

$$C = 1101_2 = 13_{10}$$



Пример 1.16

Сложить двоичные числа $A = -1010$ и $B = -0011$ в обратном и дополнительном кодах:

$$\begin{array}{r} + [A]_{OB} = 1,0101 \\ + [B]_{OB} = 1,1100 \\ \hline \end{array}$$

1,0001

└─ +1 ──>

$$[C]_{OB} = 1,0010$$

$$C = -1101_2 = -13_{10}.$$

$$\begin{array}{r} + [A]_Д = 1,0110 \\ + [B]_Д = 1,1101 \\ \hline \end{array}$$

$$[C]_Д = 1,0011;$$

$$C = -1101_2 = -13_{10}$$



При сложении чисел одного знака возможно переполнение разрядной сетки, признаком чего является несовпадение знака результата со знаками операндов. В АЛУ имеются специальные логические схемы, которые автоматически формируют признак переполнения.

Пример 1.17

Сложить двоичные числа $A = 1011$ и $B = 0111$ в обратном и дополнительном кодах:

$$\begin{array}{r} [A]_{\text{об}} = 0,1011 \\ + [B]_{\text{об}} = 0,0111 \\ \hline [C]_{\text{об}} \neq 1,0010; \end{array}$$

$$\begin{array}{r} [A]_{\text{д}} = 0,1011 \\ + [B]_{\text{д}} = 0,0111 \\ \hline [C]_{\text{д}} \neq 1,0010. \end{array}$$

В этом примере сложение положительных чисел привело к положительному переполнению: операнды — положительные, результат — отрицательный.

Пример 1.18

Сложить двоичные числа $A = -1011$ и $B = -0111$ в обратном и дополнительном кодах:

$$\begin{array}{r} [A]_{\text{об}} = 1,0100 \\ + [B]_{\text{об}} = 1,1000 \\ \hline [C]_{\text{об}} \neq 0,1101; \end{array}$$

$$\begin{array}{r} [A]_{\text{д}} = 1,0101 \\ + [B]_{\text{д}} = 1,1001 \\ \hline [C]_{\text{д}} \neq 0,1110. \end{array}$$



В этом примере сложение отрицательных чисел привело к отрицательному переполнению; операнды – отрицательные, результат – положительный.

Для упрощения обнаружения переполнения разрядной сетки используются модифицированные коды, для которых знаковый разряд в сумматоре дублируется. Положительному переполнению в знаковых разрядах соответствуют цифры 01, а отрицательному – 10. Значение знаковых разрядов 00 соответствует правильному положительному результату, а цифры 11 – отрицательному.



ПРИМЕР 1.19

Сложить двоичные числа $A = -1010$ и $B = 1100$ в обратном и дополнительном модифицированных кодах:

$$\begin{array}{r} [A]_{ОБР}^M = 11,0101 \\ + [B]_{ОБР}^M = \underline{00,1100} \\ \hline 100,0001 \\ \downarrow - \rightarrow +1 \end{array}$$

$$[C]_{ОБР}^M = 00,0010$$

$$C = 0010_2 = 2_{10}$$

$$\begin{array}{r} [A]_{ДОП}^M = 11,0110 \\ + [B]_{ДОП}^M = \underline{00,1100} \\ \hline [C]_{ДОП}^M = 00,0010 \end{array}$$

$$C = 0010_2 = 2_{10}$$



ПРИМЕР 1.20

Сложение чисел $A = -1101$ и $B = -1010$ в обратном и дополнительном модифицированных кодах:

$$\begin{array}{r} [A]_{OBR}^M = 11,0010 \\ + [B]_{OBR}^M = \underline{11,1010} \\ [C]_{OBR}^M \neq 10,1000 \end{array}$$

$$\begin{array}{r} [A]_{ДОП}^M = 11,0011 \\ + [B]_{ДОП}^M = \underline{11,0110} \\ [C]_{ДОП}^M \neq 10,1001 \end{array}$$



9 Арифметические операции умножения и деления

Операция умножения чисел состоит из k циклов, где k - число цифровых разрядов множителя. Результат умножения i -го разряда множителя на множимое называется частичным произведением, а их последовательное сложение – суммой частичных произведений (СЧП). В каждом цикле анализируется следующая цифра множителя: если это 1, то к СЧП прибавляется множимое, если 0, то сложение не выполняется. Цикл завершается сдвигом на один разряд множимого относительно СЧП или сдвигом СЧП относительно неподвижного множителя. Множимое и множитель размещаются в разрядной сетке на основе специальных схем-регистров, а СЧП – в сумматоре-регистре.

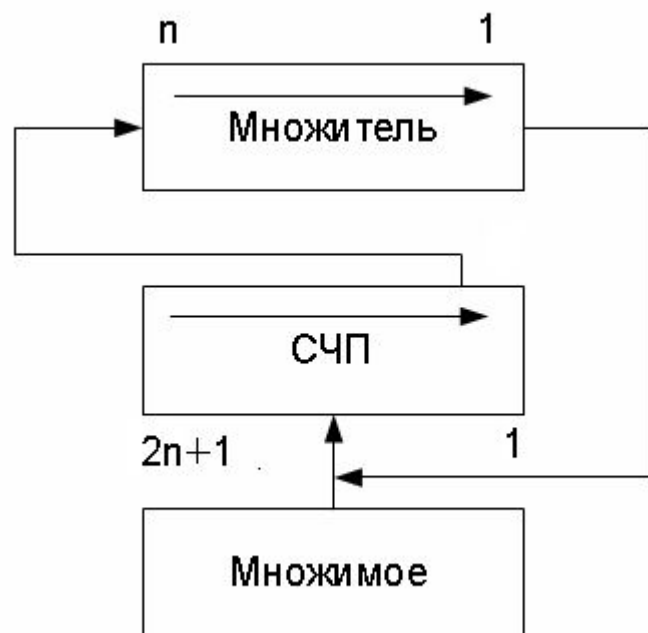


Умножение можно выполнять одним из четырех способов:

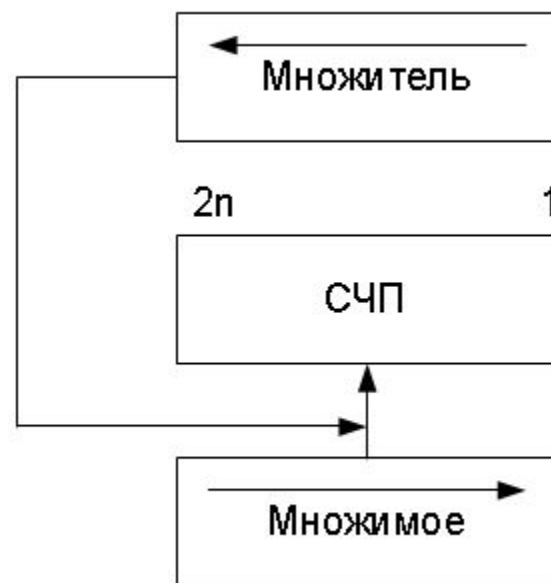
- с младших разрядов множителя со сдвигом его и СЧП вправо при неподвижном множимом (рис. 1.12, а). Старшие разряды произведения находятся в сумматоре, а младшие - в регистре множителя;
- с младших разрядов множителя при сдвиге множимого и неподвижной СЧП;
- со старших разрядов множителя при сдвиге СЧП влево и неподвижном множимом;
- со старших разрядов множителя при сдвиге его и СЧП влево (рис 1.12. б).

Если хотя бы один из сомножителей равен нулю, то произведению сразу присваивается нулевое значение.





а



б

Рис.1.12 Способы умножения :а -первый, б- четвёртый.



Алгоритм умножения целых чисел, представленных в прямом коде, на основе первого способа:

1. взять модули сомножителей и обнулить исходное значение СЧП;
2. если младшая цифра множителя равна 1, то к СЧП прибавляется множимое; если это 0, то сложение не происходит;
3. сдвиг вправо на один разряд СЧП и множителя;
4. пп. 2 и 3 последовательно выполняются k раз, после чего произведению присваивается знак, определяемый сложением по модулю 2 знаков сомножителей.



Перемножить первым способом числа $A = 13_{10} = 1101_2$ и $B = 11_{10} = 1011_2$, представленные прямыми кодами

1101	— модуль множимого
1011	— модуль множителя
0000	— исходная СЧП
<u>+1101</u>	— первое частичное произведение (ЧП)
1101	— первая СЧП
01101	— сдвиг вправо СЧП на 1 разряд
<u>+1101</u>	— второе ЧП
100111	— вторая СЧП
0100111	— сдвиг вправо СЧП на 1 разряд.
<u>+0000</u>	— третье ЧП
100111	— третья СЧП
0100111	— сдвиг вправо СЧП на 1 разряд
<u>+1101</u>	— четвертое ЧП
10001111	— четвертая СЧП
010001111	— сдвиг вправо СЧП на 1 разряд.

Произведение $C = A * B = 10001111_2 = 143_{10}$



Разработаны также методы ускоренного умножения и умножения в дополнительных кодах.

Операция деления числа ($C = A / B$) обычно сводится к последовательности вычитаний делителя B сначала из делимого A , а затем из создаваемых в процессе деления остатков R_i . Числа A и B поступают в прямом коде. В зависимости от способа вычитания делителя различают два алгоритма деления: без восстановления остатка и с восстановлением.



Для обоих алгоритмов общими является следующее:

- 1) деление осуществляется над модулями чисел A и B (при делении целых чисел в машинах длина делимого A вдвое больше длины B);
- 2) знак частного определяется сложением по модулю 2 знаков A и B ;
- 3) перед делением анализируются числа A и B на равенство нулю: если $A=0$, то частному присваивается значение 0 и операция завершается; при $B=0$ выполнение программы прерывается;
- 4) умножение частного на два осуществляется сдвигом его влево на один разряд, а операция вычитания делителя B заменяется его сложением в дополнительном коде;



5) первый остаток R_1 определяется по формуле

$$R = 2 \cdot |A| - B = 2 \cdot |A| + [-|B|]_D$$

примем, если $R_1 < 0$, то деление возможно, если $R_1 > 0$, то возникает переполнение разрядной сетки (нужно большее число разрядов);

6) цифры частного определяются, начиная со старших разрядов

$$C_i = \begin{cases} 0 & \text{при } R_i < 0; \\ 1 & \text{при } R_i \geq 0, \end{cases}$$

$i = 1, 2, \dots, k$, где k – заданное количество цифр частного. Обычно первый разряд C_i отводится для записи знака.



Алгоритм деления без восстановления остатка определяют по следующему рекуррентному соотношению:

$$R_{i+1} = \begin{cases} 2R_i + |B| & \text{при } R_i < 0; \\ 2R_i + [-|B|]_Д & \text{при } R_i \geq 0. \end{cases}$$

Алгоритм деления с восстановлением остатка определяют по следующему рекуррентному соотношению:

$$R_{i+1} = \begin{cases} 4R_{i-1} + [-|B|]_Д & \text{при } R_i < 0; \\ 2R_i + |B| & \text{при } R_i \geq 0. \end{cases}$$



Пример 1.22.

Разделить числа $A=49_{10}$ и $B=-7_{10}$ на основе алгоритма без восстановления остатка. Прямой код операндов: $[A]_{\text{пр}}=0,0110001$, $[B]_{\text{пр}}=1,111$; $[-B]_{\text{д}}=1,001$:

$$\begin{array}{r} [A]_{\text{пр}}=0,0110001 \\ |B|=0,111 \\ [-B]_{\text{д}}=1,001 \\ \hline 2|A|=0,1100010 \\ + \\ [-B]_{\text{д}}=1,001 \\ \hline R_1=1,1110010 < 0 \quad C_1=0 \\ 2R_1=1,1100100 \\ + \\ |B|=0,111 \\ \hline R_2=0,1010100 > 0 \quad C_2=1; \\ 2R_2=1,0101000 \\ + \\ [-B]_{\text{д}}=1,001 \\ \hline R_3=0,0111000 > 0 \quad C_3=1; \\ 2R_2=0,1110000 \\ + \\ [-B]_{\text{д}}=1,001 \\ \hline R_4=0,0000000 > 0 \quad C_4=1; \end{array}$$

частное $[C]_{\text{пр}}=1,111$; $C=-111=-7_{10}$

остаток равен нулю.

Знак частного $A[7] \oplus B[7] = 0 \oplus 1 = 1$.



Пример 1.23.

Разделить числа $A=49_{10}$ и $B=-7_{10}$ на основе алгоритма с восстановлением остатков:

$$[A]_{\text{ПР}}=0,0110001$$

$$|B|=0,111$$

$$[-B]_{\text{Д}}=1,001$$

$$2|A|=0,1100010$$

+

$$[-B]_{\text{Д}}=1,001$$

$$R_1=1,1110010 < 0$$

$$C_1=0;$$

$$4R_0=1,1000100$$

+

$$[-B]_{\text{Д}}=1,001$$

$$R_2=0,1010100 > 0$$

$$C_2=1;$$

$$2R_2=1,0101000$$

+

$$[-B]_{\text{Д}}=1,001$$

$$R_3=0,0111000 > 0$$

$$C_3=1;$$

$$2R_3=0,1110000$$

+

$$[-B]_{\text{Д}}=1,001$$

$$R_4=0,0000000 > 0$$

$$C_4=1;$$

$$\text{частное } [A]_{\text{ПР}}=1,111; C=-111=-7_{10}.$$

