Глава 11

АРИФМЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ КОМПЬЮТЕРОВ

11.1. Системы счисления

При знакомстве с принципами построения и работы компьютеров по аналогии с обычной школьной математикой различают двоичные арифметику и алгебру. Начнем с арифметики.

Основное понятие арифметики — это понятие числа. Числом называют абстрактное выражение количества. **Системой счисления** называют совокупность приемов построения, записи и наименования чисел.

История развития способов счета насчитывает тысячелетия. Менялись и средства счета: пальцы, камешки, узелки, счеты, арифмометры, компьютеры. Естественно было желание ученых и инженеров проектировать вычислительные устройства, работающие в привычной для нас десятичной системе. Так и происходило, пока эти устройства были механическими.

Первые электронные вычислительные машины на реле уже строились на основе двоично-десятичной системы, в которой каждая десятичная цифра кодировалась в двоичной системе. В настоящее время компьютеры работают с информацией, представленной, как правило, в двоичной системе, имеющей перед другими системами существенные преимущества.

Десятичная система

Наиболее известна широкоприменяемая на практике десятичная система. Это **позиционная** система счисления. Количество, определяемое цифрой числа, зависит от позиции этой цифры в записи числа. Например, в записи числа $A_{(10)} = 333$ одна и та же цифра 3 определяет различные количества — триста, тридцать и три. Эти количества называют **количественными эквивалентами цифр.**

К непозиционным системам относят римскую систему счисления. Например, в числе XXII количество, определяемое цифрами X и I, не зависит от их положения в записи числа.

Десятичная система имеет 10 цифр (0,1,2,...,9), что и определило название системы и ее важнейшую характеристику — **основание системы.** Обозначим его буквой p. Для десятичной системы p = 10.

Формула разложения числа по степеням основания

Пусть в десятичной системе задано некоторое число A(10) = 3745. Каждая позиция, занимаемая цифрами, называется разрядом числа. Разряды имеют названия и номера: разряд единиц, разряд десятков, разряд сотен и т.д. Названия разрядов определяют их вес: единицы, десятки, сотни, тысячи. Поэтому количественный эквивалент цифры в записи числа равен произведению значения цифры на вес разряда, где она записана.

Заданное число, не изменяя его количества, можно записать следующими способами:

$$A_{(10)} = 3745;$$
$$A_{(10)} = 3000 + 700 + 40 + 5;$$
$$A_{(10)} = 3 \cdot 1000 + 7 \cdot 100 + 4 \cdot 10 + 5;$$
$$A_{(10)} = 3 \cdot 10^3 + 7 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0.$$

Последнюю запись называют разложением числа по степеням основания. Формула разложения показывает, что число в позиционной системе можно представить в виде суммы количественных эквивалентов цифр, которые в свою очередь равны произведению цифры на степень основания, т.е. на вес разряда.

Запишем целое четырехразрядное десятичное число и формулу его разложения в общем виде:

$$A_{(10)} = a_3a_2a_1a_0 = a_3 \cdot 10^3 + a_2 \cdot 10^2 + a_1 \cdot 10 + a_0.$$

Заметим, что номера разрядов числа совпадают с показателями степени основания. Смешанная десятичная дробь, имеющая по четыре разряда в целой и дробной части, и формула его разложения запишется так (опустим знаки умножения):

$$A_{(10)} = a_3a_2a_1a_0{,}a_{-1}a_{-2}a_{-3}a_{-4} =$$

$$= a_3 10^3 + a_2 10^2 + a_1 10 + a_0 + a_{-1}10^{-1} + a_{-2}10^{-2} + a_{-3}10^{-3} + a_{-4}10^{-4}.$$

В принятых обозначениях формула количественного эквивалента цифры i-го разряда в записи десятичного числа такая: $B_i = a_i 10^i$.

Формула разложения смешанной дроби, имеющей n разрядов в целой части и m разрядов в дробной, в позиционной системе с основанием **p** запишется так:

$$A_{(p)} = a_{n-1}\ldots a_1a_0{,}a_{-1}a_{-2}\ldots a_{-m} =$$

$$= a_{n-1}p^{n-1} + \ldots + a_1p + a_0 + a_{-1}p^{-1} + a_{-2}p^{-2}\ldots + a_{-m}p^{-m}.$$

11.2. Системы счисления, применяемые в компьютерах

Ввод информации в компьютер и вывод из него результатов вычислений производится, как правило, в привычной для нас десятичной системе. Хранение и преобразование информации современные компьютеры выполняют в двоичной системе (p = 2). В качестве вспомогательных используется восьмиричная (p = 8) и шестнадцатиричная (p = 16) системы.

Двоичная система счисления

Основание двоичной системы p = 2 определяет и число цифр: 0 и 1. Формула разложения смешанной двоичной дроби по степеням основания для двоичной системы следующая:

$$A_{(2)} = a_3a_2a_1a_0{,}a^{-1}a^{-2}a^{-3}a^{-4} =$$

$$= a_3 2^3 + a_2 2^2 + a_1 2 + a_0 + a_{-1}2^{-1} + a_{-2}2^{-2} + a_{-3}2^{-3} + a_{-4}2^{-4}.$$

В двоичной системе весовой коэффициент цифры 1 в любом разряде числа совпадает со степенью основания этого разряда. Поэтому для перевода двоичного числа в десятичную систему достаточно просуммировать веса единичных разрядов. Например, если $A_{(2)} = 101$, то для перевода нужно получить сумму $A_{(10)} = 1 \cdot 2^2 + 0 \cdot 2 + 1 = 4 + 0 + 1 = 5$.

При первом знакомстве с двоичной, восьмеричной и шестнадцатеричной системами существенно помогает табл. 11.1 соответствия записей числа в различных системах.

Таблица 11.1

p = 10	p = 2	p = 8	p = 16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Восьмеричная и шестнадцатеричная системы счисления

Основание восьмеричной системы p = 8. Цифры: 0, 1, 2, 3, 4, 5, 6 и 7. Основание шестнадцатеричной системы p = 16. Цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Формулы разложения в этих системах смешанной дроби, имеющей по четыре разряда в целой и дробной части, следующие:

$$A_{(8)} = a_3 8^3 + a_2 8^2 + a_1 8 + a_0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} + a_{-3} 8^{-3} + a_{-4} 8^{-4},$$

$$A_{(16)} = a_3 16^3 + a_2 16^2 + a_1 16 + a_0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + a^{-3} 16^{-3} + a_{-4} 16^{-4}.$$

11.3. Перевод чисел из одной системы в другую

Перевод с использованием формулы разложения

Наиболее простой способ перевода заключается в суммировании количественных эквивалентов цифр заданного числа. Действия при переводе выполняются в новой системе, поэтому способ удобно использовать для перевода чисел в десятичную систему. В дальнейшем будем применять его при проверках правильности перевода чисел другими способами.

В основе способа лежит использование значений степеней основания чисел. Некоторые степени оснований 10, 2, 8 и 16 приведены в табл. 11.2. Часть клеток таблицы не заполнена ввиду значительной величины чисел.

Таблица степеней оснований p^n — Таблица 11.2

n	7	6	5	4	3	2	1	0	−1	−2	−3	−4
p=10	10^7	10^6	10^5	10^4	10^3	100	10	1	.1	.01	.001	
p=2	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16
p=8					512	64	8	1	1/8	1/64		
p=16						256	16	1	1/16			

Пример 1.

Дано: $A_{(2)} = 1101$. *Найти:* $A_{(10)}$.

Решение: Записываем формулу разложения двоичного числа по степеням основания:

$$A_{(2)} = a_3 a_2 a_1 a_0 = a_{3ш} \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2 + a_0 = a_3 \cdot 8 + a_2 \cdot 4 + a_1 \cdot 2 + a_0.$$

Подставим в формулу значения разрядов заданного двоичного числа и выполним действия:

$$A_{(10)} = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 = 13.$$

Заглянув в табл. 11.1, убедимся, что получен правильный результат. Действительно, двоичному числу $1101_{(2)}$ соответствует десятичное число $13_{(10)}$.

Ответ: $A_{(10)} = 13$.

Пример 2.

Дано: $A_{(2)} = 10111{,}101$. *Найти:* $A_{(10)}$.

Решение: Запишем формулу разложения двоичного числа:

$$A_{(2)} = a_4 a_3 a_2 a_1 a_0, a_{-1} a_{-2} a_{-3} = a_4 \cdot 16 + a_3 \cdot 8 + a_2 \cdot 4 + a_1 \cdot 2 + a_0 + a_{-1} \cdot \frac{1}{2} + a_{-2} \cdot \frac{1}{4} + a_{-3} \cdot \frac{1}{8}.$$

Подставляем в формулу значения двоичных разрядов и выполняем действия:

$$A_{(10)} = 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 + 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} = 23\frac{5}{8}.$$

Записи при переводе можно значительно уменьшить, если непосредственно суммировать степени основания, соответствующие единичным значениям разрядов заданного числа (сколько единиц в двоичной записи числа — столько слагаемых):

$$A_{(10)} = 16 + 4 + 2 + 1 + \frac{1}{2} + \frac{1}{8} = 23\frac{5}{8}.$$

Ответ: $A_{(10)} = 23\frac{5}{8}$.

Пример 3.

Дано: $B_{(8)} = 135{,}42$. *Найти:* $B_{(10)}$.

Решение: Формула разложения числа в восьмеричной системе следующая:

$$B_{(8)} = b_2 b_1 b_0, b_{-1} b_{-2} = b_2 \cdot 8^2 + b_1 \cdot 8 + b_0 + b_{-1} \cdot 8^{-1} + b_{-2} \cdot 8^{-2}.$$

Подставляем в формулу значения разрядов заданного восьмеричного числа:

$$B_{(10)} = 1 \cdot 64 + 3 \cdot 8 + 5 + 4 \cdot \frac{1}{8} + 2 \cdot \frac{1}{64} = 93\frac{34}{64}.$$

Ответ: $B_{(10)} = 93\frac{34}{64}$.

Пример 4.

Дано: $C_{(16)} = 2A{,}3E$. *Найти:* $C_{(10)}$.

Решение: Формула разложения числа в шестнадцатеричной системе:

$$C_{(16)} = c_1 c_0, c_{-1} c_{-2} = c_1 \cdot 16 + c_0 + c_{-1} \cdot \frac{1}{16} + c_{-2} \cdot \frac{1}{256}.$$

Подставляем значения разрядов заданного числа:

$$C_{(10)} = 2 \cdot 16 + 10 + 3 \cdot \frac{1}{16} + 14 \cdot \frac{1}{256} = 42\frac{62}{256}.$$

Ответ: $C_{(10)} = 42\frac{62}{256}$.

Следующая программа реализует алгоритм перевода, основанный на сложении количественных эквивалентов разрядов заданного числа. Программа легко трансформируется для перевода восьмеричных чисел в десятичную систему.

Программа 11.1

```
REM Перевод целого A(2) в A(10).
CLS
PRINT "Введи двоичное целое"
INPUT "a(2) = ", a                      'предложение ввода
```



```
DO
   b = b + (a MOD 10)*2 ^ k             'формирование результата
   k = k + 1                            'перебор номеров разрядов
   a = a \ 10                           'отбрасывание учтенных разрядов
LOOP WHILE a>0                          'проверка окончания перевода
PRINT "a(10) = "; b
END
```

При переводе чисел с использованием формулы разложения числа все действия выполняются в новой системе. В компьютерах этот метод находит применение при переводе чисел из десятичной системы в «привычную» для компьютера двоичную систему.

Перевод целых чисел делением на основание новой системы

Алгоритм перевода чисел заключается в следующем. Сначала исходное число, затем получающиеся частные делим на основание новой системы. Действия выполняем в старой системе. Записываем последнее частное и остатки в порядке обратном получения. Полученное число является записью заданного числа в новой системе.

Пример 5 наглядно показывает работу алгоритма перевода целых чисел.

Пример 5

Дано: $A_{(10)} = 35$. *Найти:* $A_{(2)}$.

Решение:

```
_35 |2
 34 _17 |2
  1  16 _8 |2
      1  8 _4 |2
         0  4 _2 |2
            0  2 |1
               0
```

Ответ: $A_{(2)} = 100011$.

Проверку правильности полученного результата удобно выполнить, используя формулу разложения числа по степеням основания:

$$A_{(10)} = 1\cdot2^5 + 0\cdot2^4 + 0\cdot2^3 + 0\cdot2^2 + 1\cdot2 + 1 =$$
$$= 32 + 0 + 0 + 0 + 2 + 1 = 32 + 2 + 1 = 35.$$

Так как действия выполняются в старой системе, рассматриваемое правило удобно использовать при переводе чисел из десятичной системы.

В примерах 6 и 7 выполняется перевод чисел из десятичной системы соответственно в восьмеричную и шестнадцатеричную системы. Проверку правильности полученных результатов легко выполнить обратным переводом с использованием формулы разложения.

Пример 6.

Дано: $A_{(10)} = 95$.
Найти: $A_{(8)}$.
Решение:

```
_95 |8
 88 _11 |8
  7   8  1
      3
```

Ответ: $A_{(8)} = 137$.

Пример 7.

Дано: $A_{(10)} = 45$.
Найти: $A_{(16)}$.
Решение:

```
_45 |16
 32   2
 13
```

Ответ: $A_{(16)} = 2D$.

Возможны другие схемы построения алгоритма перевода, например, с использованием для определения частных операции целочисленного деления (знак операции — \ ) и определением остатков с помощью функции X MOD Y. Такая схема показана в примере 8 и использована в программе 11.2.

Пример 8.

Дано: $A_{(10)} = 35$. *Найти:* $A_{(2)}$.
Решение:

35 \ 2 = 17,	35 MOD 2 = 1;
17 \ 2 = 8,	17 MOD 2 = 1;
8 \ 2 = 4,	8 MOD 2 = 0;
4 \ 2 = 2,	4 MOD 2 = 0;
2 \ 2 = 1,	2 MOD 2 = 0;
1 \ 2 = 0,	1 MOD 2 = 1.

Проверка: $A_{(10)} = 32+2+1 = 35$.
Ответ: $A_{(2)} = 100011$.

Программа 11.2

```
REM Перевод 10 — 2
CLS
INPUT "Введи десятичное число"; a
b = a                          'сохранение а для печати
k = 0                          'номера двоичных разрядов
DO
   r = a MOD 2                 'определение остатков
   a = a \ 2                   'определение частных
   c = r * 10 ^ k + c          'формирование результата
   k = k + 1
LOOP WHILE a > 0
PRINT "A(10) = "; b; "равно A(2) = "; c
END
```

Перевод правильных дробей умножением на основание новой системы

Алгоритм перевода следующий. Последовательно умножаем сначала исходное число, затем дробные части получаемых произведений на основание новой системы. При этом в целую часть будут выходить цифры записи числа в новой системе. Действия выполняются в старой системе.

Пример 9.

Дано: $A_{(10)} = 0{,}375$. *Найти:* $A_{(2)}$.
Решение:

```
 0 375
×    2
 0 750
×    2
 1 500
×    2
 1 000
```

Ответ: $A_{(2)} = 0{,}011$.

В примере 9 в целые части произведений вышли цифры 011. Можно предположить, что результат равен $A_{(2)} = 0{,}011$. Проверим полученный результат обратным переводом:

$$0{,}011_{(2)} = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}_{(10)} = 0{,}375_{(10)}.$$

В рассмотренном примере перевод заканчивается на третьем шаге, так как дробная часть становится равной нулю. Это бывает далеко не всегда.

Поставим вопросы. Когда нужно заканчивать умножение на основание новой системы? Сколько нужно получать значащих разрядов результата перевода?

Можно принять естественное правило: точность записи числа в новой системе должна быть не ниже точности числа в старой системе. Например, если задано число $C_{(10)} = 0{,}6$, то есть с точностью до 1/10, то запись этого числа в двоичной системе должна иметь четыре разряда дробной части. Это обеспечит точность до 1/16 (три разряда мало, так как третий разряд имеет вес 1/8 < 1/10). Если же исходное десятичное число задано, например, с точностью до 1/100, то принятое правило требует получения в двоичной системе семи разрядов после запятой ($2^{-7} = 1/128$), в восьмеричной системе — трех разрядов ($8^{-3} = 1/512$), в шестнадцатеричной системе — двух разрядов ($16^{-2} = 1/256$). При вычислениях следует использовать правила округления результатов. Другим ограничением может служить заранее заданное количество разрядов для записи числа в новой системе.

Пример 10.

Дано: $B_{(10)} = 0{,}67$. *Найти:* $B_{(2)}$.

Решение:

$$\begin{array}{rr}
\times\begin{array}{r} 0,\ 67 \\ 2 \\ \hline 1\ \ 34 \end{array} & \times\begin{array}{r} 0\ \ 72 \\ 2 \\ \hline 1\ \ 44 \end{array} \\
\times\begin{array}{r} 2 \\ \hline 0\ \ 68 \end{array} & \times\begin{array}{r} 2 \\ \hline 0\ \ 88 \end{array} \\
\times\begin{array}{r} 2 \\ \hline 1\ \ 36 \end{array} & \times\begin{array}{r} 2 \\ \hline 1\ \ 76 \end{array} \\
\times\begin{array}{r} 2 \\ \hline \end{array} & \times\begin{array}{r} 2 \\ \hline 1\ \ 52 \end{array}
\end{array}$$

$B_{(2)} = 0{,}10101011.$

Ответ: $B_{(2)} = 0{,}101011.$

В примере 10, записав последовательно целые части произведений, получаем следующий результат: $B_{(2)} = 0{,}10101011$. Вычислено восемь двоичных разрядов. Так как исходное число задано с точностью 1/100, а $2^{-8} = 1/256$, то получен один лишний разряд, который используем для округления результата.

Выполним проверку. Переведем полученный результат в десятичную систему:

$$B_{(10)} = 5 + \frac{1}{8} + \frac{1}{32} + \frac{1}{64} = \frac{43}{64} = 0{,}672 \approx 0{,}67.$$

Лишние 0,002 получены в результате округления в большую сторону.

В заключение рассмотрим схему перевода правильной дроби. Будем использовать рассмотренное выше правило умножения дробных частей получающихся произведений на основание новой системы. Получающиеся в каждом шаге цифра записи числа в двоичной системе в примере 11 подчеркнуты.

Пример 11.

Дано: $C_{(10)} = 9/11$. *Найти:* $C_{(2)}$.

Решение:

$$\underline{0}\frac{9}{11}\cdot 2 \to \frac{18}{11} \to \underline{1}\frac{7}{11}\cdot 2 \to \frac{14}{11} \to \underline{1}\frac{3}{11}\cdot 2 \to \frac{6}{11} \to$$

$$\to \underline{0}\frac{6}{11}\cdot 2 \to \frac{12}{11} \to \underline{1}\frac{1}{11}\cdot 2 \to \ldots$$

Выписав подчеркнутые цифры, получаем результат $C_{(2)} = 0{,}1101$ с недостатком. Это число $C_{(10)} = 13/16$. Перевод исходного числа (9/11) и результата (13/16) в десятичную дробь дает результаты: 0,818 и 0,813.

Смешанная дробь переводится в новую систему счисления по частям: целая часть — методом деления, дробная часть — методом умножения на основание новой системы.

Поразрядные способы перевода

Перевод чисел существенно упрощается, если основания старой (p) и новой (q) систем связаны соотношением $p = q^k$ или $p^k = q$, где k — целое число. Это, например, системы восьмеричная и двоичная (k = 3), шестнадцатеричная и двоичная (k = 4), девятеричная и троичная (k = 3) и т.д.

Рассмотрим на примерах переводы чисел из восьмеричной и шестнадцатеричной систем в двоичную и обратно.

Пример 12.

Дано: $A_{(8)} = 132{,}52$. *Найти:* $A_{(2)}$.

Решение.

Для получения результата нужно каждую восьмеричную цифру заданного числа записать тремя двоичными разрядами — триадой (k = 3):

$A_{(8)} = 1\quad 3\quad 2,\quad 5\quad 2;$

$A_{(2)} = 001\ \ 011\ \ 010,\ 101\ \ 010;$

Ответ: $A_{(2)} = 1011010{,}10101$.

Пример 13.

Дано: $B_{(16)} = 20E{,}B8$. *Найти:* $B_{(2)}$.

Решение.

В этом примере каждая шестнадцатеричная цифра записывается четырьмя двоичными разрядами — тетрадой (k = 4):

$B_{(16)} = 2\quad 0\quad E,\quad B\quad 8;$

$B_{(2)} = 0010\ \ 0000\ \ 1110,\ 1011\ \ 1000;$

Ответ: $B_{(2)} = 1000001110{,}10111$.

Пример 14.

Дано: $C_{(2)} = 11001111{,}00011$. *Найти:* $C_{(8)}$ и $C_{(16)}$.

Решение.

Для перевода нужно разбить заданное двоичное число влево и вправо от запятой на триады (тетрады), при необходимости дополняя их нулями. Затем каждую триаду (тетраду) записать цифрами восьмеричной (шестнадцатеричной) системы:

$C_{(2)} = 11001111{,}00011;$	$C_{(2)} = 11001111{,}00011;$
$C_{(2)} = 011\ 001\ 111,\ 000\ 110;$	$C_{(2)} = 1100\,1111,\ 0001\ 1000;$
$C_{(8)} = 3\quad 1\quad 7,\quad 0\quad 6;$	$C_{(16)} = B\quad F,\quad 1\quad 8;$
Ответ: $C_{(8)} = 317{,}06$.	Ответ: $C_{(16)} = BF{,}18$.

Поразрядные способы перевода чисел можно использовать для сокращения действий при переводе числа, например, из десятичной системы в двоичную. Для этого целое число делением (дробное — умножением) сначала переводят в восьмеричную систему, а затем из восьмеричной системы поразрядно в двоичную систему.

Если в качестве промежуточной системы использовать двоичную, то существенно упрощается перевод из восьмеричной системы в шестнадцатеричную и обратно. Это показано в следующем примере.

Пример 15.

Дано: $A_{(8)} = 275{,}034$. *Найти:* $A_{(16)}$.

Решение:

$A_{(8)} = 2\quad 7\quad 5,\quad 0\quad 3\quad 4;$

$A_{(2)} = 010\ \ 111\ \ 101,\ 000\ \ 011\ \ 100;$

$A_{(2)} = 1011\ 1101,\ 0000\ 1110;$

$A_{(16)} = B\quad D,\quad 0\quad E;$

Ответ: $A_{(16)} = BD{,}0E$.

Быстрый способ перевода, использующий устный счет

Записав единицу, приписываем к ней справа нули (1, 10, 100, 1000, 10000,...) и переводим в десятичную систему. Получаем числа 1, 2, 4, 8, 16,... . С приписыванием справа нуля двоичное число увеличивается вдвое. Если же приписать единицу, то число увеличится вдвое плюс единица.

Пример 16.

Дано: $A_{(2)} = 1010011{,}100101$. *Найти:* $A_{(10)}$.

Решение.

Последовательно открывая разряды целой части числа, получаем:

$1,\ 2,\ 4 + 1 = 5,\ 10,\ 20,\ 40 + 1 = 41,\ 82 + 1 = 83.$

С дробной частью поступаем так же:

1, 2, 4, 8 + 1 = 9, 18, 36 + 1 = 37.

Шестой разряд после запятой имеет вес $2^6 = 64$. Поэтому дробная часть равна 37/64.

Ответ: $A_{(10)} = 83$.

11.4. Арифметические действия в двоичной системе

Все правила выполнения арифметических действий в любой позиционной системе одинаковы и совпадают поэтому с правилами для десятичной системы. Рассмотрим сложение и умножение чисел в двоичной системе. Таблицы сложения (табл. 11.3) и умножения (табл. 11.4) для двоичной системы предельно просты.

Таблица 11.3
Сложение чисел в двоичной системе

	0	1
0	0	1
1	1	1

Таблица 11.4
Умножение чисел в двоичной системе

	0	1
0	0	0
1	0	1

При сложении чисел необходимо правильно формировать переносы в старшие разряды с учетом переносов из младших разрядов чисел. При умножении суммируются частичные произведения (произведения множимого на па цифры разрядов множителя). Заметим, что в двоичной системе частичные произведения формируются очень просто: они равны или сдвинутому влево множимому, если цифра множителя равна единице, или нулю, если цифра множимого равна нулю.

Рассмотрим простейшие примеры сложения и умножения двоичных чисел. Параллельно будем выполнять те же действия в десятичной системе. Это поможет контролировать правильность результата.

Пример 17.

Дано: $A = 1010_{(2)} = 10_{(10)}$;
$B = 111_{(2)} = 7_{(10)}$.

Найти: $C_{(2)} = A_{(2)} + B_{(2)}$.

Решение:

$$+\begin{array}{r} A_{(2)} = 1010 \\ B_{(2)} = 111 \\ \hline C_{(2)} = 10001 \end{array} \qquad +\begin{array}{r} A_{(10)} = 10 \\ B_{(10)} = 7 \\ \hline C_{(10)} = 17 \end{array}$$

Ответ: $C_{(2)} = 10001$

Пример 18.

Дано: $A = 1101_{(2)} = 13_{(10)}$.
$B = 1011_{(2)} = 11_{(10)}$.

Найти: $C_{(2)} = A_{(2)} \cdot B_{(2)}$.

Решение:

$$\times\begin{array}{r} A_{(2)} = 1101 \\ B_{(2)} = 1011 \\ \hline 1101 \\ +\ \ 1101 \\ 0000 \\ 1101 \\ \hline C_{(2)} = 10001111 \end{array} \qquad \times\begin{array}{r} A_{(10)} = 13 \\ B_{(10)} = 11 \\ \hline C_{(10)} = 143 \end{array}$$

11.5. Сравнение систем счисления

На первый взгляд вне конкуренции хорошо знакомая нам десятичная система. Однако она используется только при вводе информации в компьютер и выводе из него. Это обусловлено следующими причинами.

При хранении и передаче информации каждую цифру необходимо представлять некоторой физической величиной, например, амплитудой напряжения, тока, направлением намагниченности магнитного материала и т.п. В условиях помех, чем больше число градаций этих физических величин (для десятичной системы 10), тем больше вероятность переходов с одной градации к другой и появления ошибок. Это приводит к уменьшению надежности хранения и передачи информации. Возможность появления таких ошибок минимальна при использовании двоичной системы.

При кодировании информации в двоичной системе наиболее просто технологически реализуются электронные схемы, выполняющие операции над числами (транзистор открыт или закрыт, импульс тока есть или нет, участок поверхности магнитного диска намагничен или размагничен). К тому же и действия над двоичными числами, как было показано, выполняются весьма просто.

К недостаткам двоичной системы следует отнести необходимость и трудоемкость перевода чисел из десятичной системы при вводе информации в компьютер и в десятичную систему при выводе результатов. Отметим также, что двоичная система самая неэкономная по записи чисел. Она требует больше разрядов, чем запись того же числа в других системах.

Задания для самостоятельной работы

Перевести числа:

a) Дано: $A_{(2)}$ = 101011,000111. Найти: $A_{(8)}$, $A_{(10)}$, $A_{(16)}$.
b) Дано: $B_{(8)}$ = 150,74. Найти: $B_{(2)}$, $B_{(10)}$, $B_{(16)}$.
c) Дано: $C_{(10)}$ = 87,29. Найти: $C_{(2)}$, $C_{(8)}$, $C_{(16)}$.
d) Дано: $D_{(16)}$ = A0,F8. Найти: $D_{(2)}$, $D_{(8)}$, $D_{(10)}$.
e) Дано: $E_{(10)}$ = 7/11. Найти: $E_{(10)}$.
f) Дано: $F_{(5)}$ = 304,12. Найти: $F_{(10)}$.
g) Дано: $G_{(10)}$ = 47,8. Найти: $G_{(3)}$.
h) Дано: $H_{(р.с.с.)}$ = MDCCCXII. Найти: $H_{(10)}$.

Выполнить действия:

k) Дано: $A_{(2)}$ = 10101, $B_{(2)}$ = 111. Найти: $C_{(2)} = A_{(2)} + B_{(2)}$.
l) Дано: $D_{(2)}$ = 1101, $E_{(2)}$ = 1011. Найти: $F_{(2)} = D_{(2)} \cdot E_{(2)}$.

11.6. Способы представления чисел в компьютере

Желание обойтись в процессоре только сумматором привело к замене вычитания чисел сложением специальных кодов чисел. Поясним это простейшим примером действий с числами в десятичной системе.

Пусть требуется выполнить действие 74 – 48. Код положительного числа равен самому числу. Код отрицательного числа формируется как дополнение этого числа до 100: 100 – 48 = 52. В компьютере это выполняется очень просто, без применения вычитания.

При сложении кодов чисел получаем результат 74 + 52 = 126. При этом возникает перенос (одна сотня) из старшего разряда чисел (переполнение). Этот перенос отбрасывается, устраняется схемным путем. Результат вычитания равен 26.

В качестве кодов используют прямой, обратный и дополнительный коды. **Машинный код** числа состоит из знакового разряда и цифровых разрядов. Принято кодировать знак плюс цифрой 0, знак минус — цифрой 1. Для отличия машинных кодов от других записей чисел используют квадратные скобки. Знаковый разряд от цифровых разрядов будем отделять точкой.

Примем для простоты примеров разрядность чисел равной пяти разрядам: один разряд для записи знака и четыре разряда для записи цифровых разрядов числа. Рассмотрим примеры записи чисел в виде машинных кодов. В качестве чисел будем брать числа $1 > A \geq 1/16$, то есть правильные дроби.

Прямой код

Формируется записью в знаковый разряд знака числа, в цифровые разряды — значений цифровых разрядов числа.

Пример 19.

Дано: $A_{(10)} = \dfrac{12}{16}$, $B_{(10)} = -\dfrac{5}{16}$.

Найти: запись A и B в прямом коде.

Решение:

$A_{(2)} = +0{,}1100$; $B_{(2)} = -0{,}0101$;

Ответ: $[A]_{пк} = 0.1100$; $[B]_{пк} = 1.0101$.

Обратный код

Обратный код положительных чисел равен их прямому коду. Обратный код отрицательных чисел формируется из прямого и прямой код из обратного инверсией (изменением на обратное, противоположное значение) цифровых разрядов числа.

Пример 20.

Дано: $A_{(10)} = \frac{5}{16}$, $B_{(10)} = -\frac{13}{16}$.

Найти: запись A и B в прямом и обратном коде.

Решение:

$A_{(2)} = +0{,}0101$; $B_{(2)} = -0{,}1101$;

Ответ: $[A]_{пк} = 0.0101$; $[B]_{пк} = 1.1101$;

$[A]_{ок} = 0.0101$; $[B]_{ок} = 1.0010$.

Дополнительный код

Донолнительный код положительных чисел равен их прямому коду. Дополнитсльпый код отрицательных чисел формируется как дополнение заданиого числа до единицы. Дополнительный код удобно получать из обратного кода прибавлением единицы к младшему цифровому разряду.

Пример 21.

Дано: $A_{(10)} = \frac{5}{16}$, $B_{(10)} = -\frac{13}{16}$.

Найти: $[A]_{пк}$, $[A]_{ок}$, $[A]_{дк}$, $[B]_{пк}$, $[B]_{ок}$, $[B]_{дк}$.

Решение:

$A_{(2)} = +0{,}0101$; $B_{(2)} = -0{,}1101$;

Ответ:
$[A]_{пк} = 0.0101$; $[B]_{пк} = 1.1101$;
$[A]_{ок} = 0.0101$; $[B]_{ок} = 1.0010$;
$[A]_{дк} = 0.0101$; $[B]_{дк} = 1.0011$.

Выполнение арифметических операций

При выполнении онераций со знаковыми разрядами кодов чисел действуют как с цифровыми. При действиях в обратном коде перенос из знакового разряда циклически нереносится к младшему цифровому разряду и суммируется с ним, а в донолнительном коде перенос из знакового разряда не учитывается и отбрасывается.

Результаты выполнения действий в примерах будем переводить в прямой код. При положительпом результате $[A]_{ок} = [A]_{дк} = [A]_{пк}$. Прямой код отрицательных чисел из обратного кода получаем инверсией цифровых разрядов. Прямой код отрицательных чисел из дополнительного получаем инверсией значений цифровых разрядов и нрибввлением единицы младшего разряда.

Пример 22 .

Дано: $A_{(10)} = \frac{5}{16}$, $B_{(10)} = -\frac{12}{16}$.

Найти: C = A+B, действия выполнить в кодах.

Решение:

$A_{(2)} = 0{,}0101$; $[A]_{пк} = 0.0101$;

$B_{(2)} = -0{,}1100$; $[B]_{пк} = 1.1100$;

$$+\begin{array}{l}[A]_{ок} = 0.0101;\\ [B]_{ок} = 1.0011;\\ \hline\end{array} \qquad +\begin{array}{l}[A]_{дк} = 0.0101;\\ [B]_{дк} = 1.0100;\\ \hline\end{array}$$

Ответы:

$$\begin{array}{ll}[C]_{ок} = 1.1000; & [C]_{дк} = 1.1001;\\ [C]_{пк} = 1.0111; & [C]_{пк} = 1.0111;\\ C_{(2)} = -0{,}0111; & C_{(2)} = -0{,}0111;\\ C_{(10)} = -\frac{7}{16}; & C_{(10)} = -\frac{7}{16}.\end{array}$$

Пример 23.

Дано: $A_{(10)} = \frac{11}{16}$, $B_{(10)} = \frac{3}{16}$.

Найти: C = A − B, действия вынолнить в обратпом и донолнительном кодах.

Решение:

$$\begin{array}{lll}[A]_{пк} = 0.1011; & [A]_{ок} = 0.1011; & [A]_{дк} = 0.1011;\\ [-B]_{пк} = 1.0011; & +[-B]_{ок} = 1.1100; & +[-B]_{дк} = 1.1101;\\ \hline & 10.0111 & 10.1000\\ & +\quad 1 & \\ \hline\end{array}$$

Ответы:

$$\begin{array}{ll}[C]_{ок} = 0.1000 & [C]_{дк} = 0.1000\\ C_{(10)} = \frac{8}{16}; & C_{(10)} = \frac{8}{16}.\end{array}$$

Для получения ответа в обратном коде перенос из старшего разряда суммируется с младшим разрядом результата. В дополнительном коде такой перенос отбрасывается.

Примеры показывают, что действия над числами в кодах приводят к получению правильных результатов, т.е. **сумма кодов дает код суммы.**

Переполнение и машинные нули

Одна из важнейших характеристик компьютера — разрядность обрабатываемых им двоичных чисел. Количество разрядов, отводимых в компьютере для хранения, передачи и обработки двоичных чисел называют **разрядной сеткой компьютера.** Первые процессоры на интегральных схемах были четырехразрядными. В настоящее время широко распространены 16-разрядные и 32-разрядные процессоры. Появились 64-разрядные процессоры. Чем больше разрядность, тем больше диапазон обрабатываемых чисел, тем выше точность вычислений.

Если результат выполнения операции выходит за диапазон представления чисел и не может быть записан в заданном числе разрядов, то такую ситуацию называют **переполнением разрядной сетки.** С переполнением (Overflow) нередко имеем дело при решении на компьютере задач вычислительного характера.

Если же результат настолько мал, что не может быть представлен в заданной разрядной сетке, то компьютером в дальнейшем он принимается равным нулю. Ситуацию, когда число не равно нулю, но принимается компьютером за нуль, называют появлением **машинного нуля.**

Рассмотрим примеры.

Пример 24.

Дано: $A_{(10)} = \frac{11}{16}$; $B_{(10)} = \frac{9}{16}$.

Найти: C = A + B. Действия выполнить в обратном коде.

Решение:

$A_{(2)} = 0{,}1011$; $[A]_{ок} = 0.1011$; $B_{(2)} = 0{,}1001$; $[B]_{ок} = 0.1001$;

$$\text{а)}\; +\begin{array}{l} [A]_{ок} = 0.1011; \\ [B]_{ок} = 0.1001; \\ \hline [C]_{ок} = 1.0100; \end{array} \qquad \text{б)}\; +\begin{array}{l} [A]_{мок} = 00.1011; \\ [B]_{мок} = 00.1001; \\ \hline [C]_{мок} = 01.0100; \end{array}$$

Оценим результат решения примера 24,а. Суммировали два положительных числа, а получили отрицательный результат. Перенос из старшего цифрового разряда занял знаковый разряд и исказил знак результата. Произошло переполнение разрядной сетки, сумма заданных чисел больше единицы.

Для выявления подобных переполнений применяют **модифицированные коды,** отличающиеся тем, что имеют по два знаковых разряда, как показано в примере 24,б. Положительные числа в модифицированных кодах имеют в знаковых разрядах два нуля (00.), отрицательные — две единицы (11.). Признаком переполнения служит различие значений знаковых разрядов (01. или 10.). При этом первый знак указывает на знак результата.

Пример 25.

Дано: $A_{(10)} = \frac{2}{16}$, $B_{(10)} = \frac{5}{16}$.

Найти: C = A · B.

Решение.

При умножении чисел знак результата определяется по знакам сомножителей логическим путем по известным правилам. Цифровые разряды произведения определяются перемножением цифровых разрядов сомножителей.

$$A_{(2)} = 0{,}0010; \quad B_{(2)} = 0{,}0101; \qquad \times\begin{array}{r} 0{,}0010 \\ 0{,}0101 \\ \hline 0010 \\ +\;0010 \\ \hline 0{,}00001010 \end{array}$$

Ответ: $C_{(2)} = 0{,}0000$.

В результате умножения получили результат, значащие цифры которого не вписываются в разрядную сетку.

Произведение равно $0,00001010_{(2)} = 10/256_{(10)}$. В разрядную сетку записывается $C_{(2)} = 0,0000$, то есть машинный нуль.

Для уменьшения возможности появления машинных нулей нужно или увеличивать число разрядов в разрядной сетке компьютера, или вводить некоторые масштабные множители. Например, результат в примере 25 можно записать так: $C_{(2)} = 0,1010 \cdot 2^{-4}$. Использование масштабных коэффициентов реализовано в нормальной форме представления чисел.

11.7. Формы представления чисел в компьютере

Различают естественную, с фиксированной точкой (запятой) и нормальную, экспоненциальную, с плавающей точкой (запятой) формы представления чисел.

При естественной форме представления чисел точка, отделяющая разряды целой части числа от дробной части, фиксируется в определенном месте разрядной сетки, например, перед старшим цифровым разрядом, непосредственно после знакового разряда. В этом случае процессор оперирует с правильными дробями, что исключает переполнение при умножении и облегчает подбор масштабных коэффициентов.

16-разрядная сетка позволяет получить 2^{16} различных чисел без знака. Знак занимает один разряд разрядной сетки. При выполнении действий над числами возможно получение результата в виде +0 (0.0000) и −0 (1.0000). Последнее представление нуля в компьютере изменяется на +0. Двойное кодирование нуля уменьшает общее количество представляемых чисел на одно число.

Нормальную форму также называют экспоненциальной формой. Нормальная форма позволяет при одинаковой разрядности получать существенно больший диапазон представления чисел. Это приводит к уменьшению вероятности переполнения разрядной сетки и появления машинных нулей. Кроме того нормальная форма просто решает вопрое изменения значений сопровождающих числа масштабных коэффициентов.

До недавнего времени процессоры обрабатывали числа в форме с плавающей точкой по специальным микропрограммам. Затем к обычным процессорам в помощь стали добавлять так называемые математические процессоры, работающие с числами с плавающей точкой в 5-15 раз быстрее обычных. Современные процессоры обрабатывают числа, как в естественной, так и в нормальной форме.

В нормальной, экспоненциальной форме числа представляются в виде $A = Ma \cdot 2^{Pa}$, где Ma — мантисса числа, Pa — порядок.

Каждое число записывается четырьмя элементами: «знак мантиссы, он же знак числа + мантисса + знак порядка + порядок». В современных компьютерах примерное соотношение числа разрядов между этими элементами следующее: $1 + 23 + 1 + 7 = 32$ — для вещественных чисел одинарной точности и $1 + 52 + 1 + 10 = 64$ — для чисел двойной точности.

Сравним диапазоны представления чисел для естественной и нормальной формы. Для наглядности возьмем десятиразрядные числа. В естественной форме один разряд занят знаком числа и девять разрядов цифровых. Диапазон представления чисел при этом равен от $-(2^9 - 1)$ до $+(2^9 - 1)$, то есть от −511 до +511.

В нормальной форме отведем 6 разрядов под мантиссу со знаком и 4 разряда под порядок со знаком. В этом случае максимальное значение мантиссы равно $2^5 - 1 = 31$, а максимальный порядок равен $2^3 - 1 = 7$, поэтому диапазон представления чисел равен от $-31 \cdot 2^7$до $+31 \cdot 2^7$, то есть от $-31 \cdot 128$ до $+31 \cdot 128$.

Сравнение показывает, что при одинаковой разрядности диапазон чисел в экспоненциальной форме значительно больше, чем в естественной форме. Заметим, что с увеличением разрядности это различие в диапазонах увеличивается по экспоненте.

Можно ошибочно предположить, что с увеличением диапазона увеличивается и количество представляемых различных чисел. Но при заданной разрядности количество различных комбинаций нулей и единиц увеличить невозможно. Количество чисел при представлении в нормальной форме меньше, чем в естественной. Сказывается двойственное представление нуля (+0 и −0), но теперь в значениях порядка.

Если в естественной форме шаг изменения чисел постоянный и равен единице младшего разряда чисел, то экспоненциальная форма приводит к неравномерному шагу, существенно увеличивающемуся к краям диапазона.

Действия над числами в нормальной форме

Как правило, все операнды поступают из намяти для выполнения операций и результаты отправляются для хранения в намять в нормализованном виде, т.е. старший значащий разряд мантиссы записан непосредственно после знакового разряда числа. Нормализованное число представляется с максимально возможной точностью.

Пример 26.

Дано: $A_{(2)} = 11,1$; $B_{(2)} = 0,01$; $C_{(2)} = -0,1011$.

Найти: запись A, B и C в экспоненциальной форме, в нормализованном виде, в прямом коде; разрядность: 6 разрядов для мантиссы со знаком, 4 разряда для порядка.

Решение:

$A_{(2)} = 11,1 = +0,111 \cdot 10^{+010}$,

$B_{(2)} = +0,01 = +0,10 \cdot 10^{\ 001}$,

$C_{(2)} = -0,1011 \cdot 10^{1000}$.

Ответ: $[A]_{пк} = 0.11100.0.010$;

$[B]_{пк} = 0.10000.1.001$;

$[C]_{пк} = 1.10110.0.000$.

Сложение

При выполнении операций сложения и вычитания чисел с плавающей точкой необходимо предварительно выровнять порядки. При этом сдвигается вправо число, имеющее меньший порядок на количество разрядов, равное разности порядков. Рассмотрим примеры.

Пример 27.

Дано: $A_{(10)} = 2\frac{1}{4}$. $A_{(2)} = 10,01$, $B_{(10)} = -\frac{3}{8}$, $B_{(2)} = -0,011$.

Найти: C = A+B. Сложение выполнить в нормальной форме, в обратном коде.

Решение: $A_{(2)} = 10.01$, $[A]_{пк} = [A]_{ок} = 0.10010.0.010$;

$B_{(2)} = -0.011$, $[B]_{пк} = 1.11000.1.001$, $[B]_{ок} = 1.00111.1.110$.

Определяем разность порядков:

$[Pa]_{ок} = 0.010$, $[Pв]_{ок} = 1.110$, $[-Pв]_{ок} = 0.001$.

$$
\begin{array}{rl}
 & [Pa]_{ок} = 0.010 \\
+ & [-Pв]_{ок} = 0.001 \\
\hline
 & [R]_{ок} = 0.011
\end{array}
$$

Положительный результат разности порядков R указывает на то, что меньший порядок у второго слагаемого, у числа B. Величина разности показывает, на сколько разрядов необходимо для выравнивая порядков сдвинуть вправо мантиссу числа B.

Сдвинутая на три разряда вправо мантисса равна $[Mв]_{ок} = 1.11100$. При сдвиге учитываем, что в обратном коде отрицательного числа значения разрядов инвертированы, поэтому при сдвиге освобождающиеся разряды заполняются единицами (в прямом коде — нулями).

Порядки выровнены, поэтому можно выполнить сложение мантисс и приписать им больший, теперь общий для обоих слагаемых порядок. Мантиссы складываем в модифицированном коде (два знаковых разряда числа), что позволяет выявлять переполнение разрядной сетки.

$$
\begin{array}{rlr}
+ & [Ma]_{мок} = & 00.10010 \\
 & [Mв]_{мок} = & 11.11100 \\
\hline
 & & 100.01110 \\
 & + & 1 \\
\hline
 & [Mc]_{мок} = & 00.01111
\end{array}
$$

Результат сложения мантисс положителен и ненормализован. Переполнение отсутствует. Для нормализации нужно сдвинуть мантиссу результата на один разряд влево. При этом чтобы не исказить величину числа в целом, необходимо уменьшить на единицу порядок.

Получили результат: $[C]_{мок} = 00.01111.0.010$. После нормализации имеем: $[C]_{мок} = 0.11110.0.001$ (увеличиваем мантиссу — уменьшаем порядок).

Ответ: $[C]_{ок} = [C]_{пк} = 0.1111.0.001$;

$C_{(2)} = 1.111$;

$C_{(10)} = 15/16 * 21 = 1\ 7/8$.

Умножение и деление

В нормальной форме числа А и В можно записать в следующем виде: $A = Ma \cdot 2^{Pa}$ и $B = Mв \cdot 2^{Pв}$. При умножении и делении чисел А и В справедливы следующие формулы:

$$\mathbf{A \cdot B = Ma \cdot 2^{Pa} \cdot Mв \cdot 2^{Pв} = Ma \cdot Mв \cdot 2^{Pa} + Pв},$$

$$\mathbf{A/B = Ma \cdot 2^{Pa}/Mв \cdot 2^{Pв} = Ma/Mв \cdot 2^{Pa-Pв}}.$$

Для выполнения умножения (деления) чисел в экспоненциальной форме, форме с плавающей точкой нужно перемножить (разделить) мантиссы и приписать результату порядок равный сумме (разности) порядков исходных чисел.

11.8. Примеры использования других систем

Теоретически было доказано, что самой экономичной по затратам оборудования для построения компьютеров является троичная система. Когда вопросы экономии оборудования стояли очень остро (компьютеры первого и второго поколения) была построена единственная в своем роде вычислительная машина Сетунь, в которой использовалась троичная система.

Быстродействие компьютера зависит от скорости работы в процессоре сумматора чисел. В худшем случае, при сложении чисел 11...111+00...001 перенос должен пройти через все разряды сумматора. Имеются системы, в которых переносы из разряда в разряд отсутствуют. К ним относится Система счисления в Остаточных Классах (СОК). Ее еще называют системой вычетов, китайской системой (вычетами занимались математики конфуцианской школы), чешской системой (профессор Свобода первый рассматривал возможность использования СОК для построения компьютеров).

СОК — непозиционная система. Основаниями системы служат взаимно простые числа, например, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$. Диапазон представления чисел равен произведению оснований: $D = p_1 \cdot p_2 \cdot p_3 = 3 \cdot 5 \cdot 7 = 105$, т.е. при этих основаниях можно однозначно представлять числа от 0 до 104. Любое число в этом

диапазоне записывается остатками от деления этого числа на выбранные основания: $A_{(сок)} = (a_1, a_2, a_3)$, где $a_1 = A \bmod p_1$, $a_2 = A \bmod p_2$, $a_3 = A \bmod p_3$. Например, число $A_{(10)} = 13$ запишется в СОК с выбранными основаниями так: $A_{(сок)} = (1, 3, 6)$.

Действия сложение, вычитание и умножение выполняются в СОК поразрядно, по модулю оснований. Это позволяет выбирать результаты из таблиц, получать схемным путем. Рассмотрим пример сложения и умножения чисел.

Пример 28.

Дано: $A_{(10)} = 12$, $B_{(10)} = 8$. Основания СОК: 3, 5 и 7.
Найти: C = A+B, D = A(B.
Решение:

$$+\begin{array}{l} A_{(сок)} = (0, 2, 5) \\ \underline{B_{(сок)} = (2, 3, 1)} \\ C_{(сок)} = (2, 0, 6) \end{array} \qquad \times\begin{array}{l} A_{(сок)} = (0, 2, 5) \\ \underline{B_{(сок)} = (2, 3, 1)} \\ D_{(сок)} = (0, 1, 5) \end{array}$$

Правильность результата можно проверить переводом чисел 12+8 = 20 и $12 \cdot 8 = 96$ в СОК.

Ответ: $C_{(сок)} = (2, 0, 6)$, $D_{(сок)} = (0, 1, 5)$.

К сожалению, деление в системе остатков затруднено, что и определило малое распространение ее при построении компьютеров. Была построена быстродействующая специализированная ЭВМ, успешно решавшая задачи идентификации объектов при обзоре космического пространства.

Задания для самостоятельной работы

1. Заданпые числа нeревести н двоичную систему и занисать а прямом, обратном и дополнительном коде (один разряд знаковый и четыре цифроных). Выполпить сложение и вычитапие чисел в обратпом и дополнительном кодах. Проверить правильность нолучепных результатов.

   a) $A_{(10)} = \frac{11}{16}$, $B_{(10)} = -\frac{4}{16}$; b) $C_{(10)} = -\frac{6}{16}$, $D_{(10)} = -\frac{7}{16}$;

   c) $E_{(10)} = -\frac{9}{16}$, $F_{(10)} = \frac{5}{16}$; d) $G_{(10)} = \frac{13}{16}$, $H_{(10)} = \frac{2}{16}$.

2. Задапные числа перевести в двоичную систему и запиеать в прямом коде в нормальпой форме в нормализовапном виде. Для записи мантиссы со знаком отвести 6 разрядов, для записи порядка со знаком — 4 разряда. Выполнить сложение и вычитапие чисел в модифицированпых обратном и дополнительном кодах. Результат пормализовать и проверить на правильпость получения.

   a) $A_{(10)} = 7{,}5$; $B_{(10)} = 2{,}5$; b) $C_{(10)} = 3{,}25$; $D_{(10)} = -9{,}5$;

   c) $E_{(10)} = 5{,}25$; $F_{(10)} = -6{,}25$; d) $G_{(10)} = 6{,}5$; $H_{(10)} = 8{,}25$.