ГЛАВА II

АРИФМЕТИЧЕСКИЕ ОСНОВЫ

4. СИСТЕМЫ СЧИСЛЕНИЯ И ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ В МАШИНАХ

Общепринятая система записи чисел позволяет любое число представить с помощью десяти различных цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Отсюда происходит ее название «десятичная система счисления».

При записи чисел каждая цифра изменяет свое значение в зависимости от ее положения в ряду цифр, изображающих число. Например, в числе 999,9 первая слева девятка означает количество сотен, содержащихся в числе, вторая — количество десятков, третья (стоящая перед запятой) — количество единиц. Наконец девятка, стоящая после запятой, означает количество десятых долей. Другими словами, упомянутое число может быть записано в следующем виде:

$$999{,}9 = 9\cdot10^2 + 9\cdot10^1 + 9\cdot10^0 + 9\cdot10^{-1}.$$

Точно так же число 8315,25 можно записать следующим образом:

$$8305{,}25 = 8\cdot10^3 + 3\cdot10^2 + 0\cdot10^1 + 5\cdot10^\circ + 2\cdot10^{-1} + 5\cdot10^{-2}$$

и т. п.

Ввиду того что значение каждой цифры изменяется в зависимости от ее положения (позиции) в ряду цифр, изображающих число, десятичную систему счисления называют позиционной.

Число десять называют основанием системы счисления и записывают с помощью двух цифр: «10».

Десятичная позиционная система счисления не является единственной возможной позиционной системой счисления. Для записи чисел можно применять систему счисления с любым (конечно, целым) основанием.

Например, принимая за основание системы счисления число восемь, приходим к восьмеричной позиционной системе счисления. В этой системе всякое число можно представить с помощью восьми различных цифр: 0, 1, 2, 3, 4, 5, 6, 7. Основание системы восемь будет записываться с помощью двух цифр в виде «10». Число сто девяносто пять, которое в десятичной системе счисления имеет вид: $195 = 1 \cdot 10^2 + 9 \cdot 10^1 + 5 \cdot 10^\circ$ (10 означает десять), в восьмеричной системе будет иметь вид: $303 = 3 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^\circ$ (10 означает восемь).

Самое меньшее количество различных цифр необходимо для записи чисел в двоичной позиционной системе счисления. Основанием такой системы является число два и используются в ней цифры: 0 и 1. Число два записывается двумя цифрами: «10». Для примера приведем несколько чисел, записанных в двоичной системе:

0— нуль	1001 — девять
1— один	1010 — десять
10— два	1011 — одиннадцать
11— три	1100 — двенадцать
100— четыре	1101 — тринадцать
101— пять	1110 — четырнадцать
110 — шесть	1111 — пятнадцать
111 — семь	10000 — шестнадцать
1000 — восемь	

Число сто девяносто пять в двоичной системе счисления будет записано в следующем виде:

$$11000011 = 1\cdot10^{111} + 1\cdot10^{110} + 0\cdot10^{101} + 0\cdot10^{100} + 0\cdot10^{11} + 0\cdot10^{10} + 1\cdot10^1 + 1\cdot10^\circ.$$

Здесь «10» означает два. Показатели степени так же записаны в двоичной системе счисления.

Нетрудно сообразить как будут записываться числа в системах счисления с другими основаниями (например в троичной или пятеричной системе).

Если основание системы больше десяти, то цифр, применяемых в десятичной системе счисления, уже не будет хватать для записи чисел. Например в шестнадцатиричной системе счисления нужно ввести дополнительные цифры для обозначения чисел от десяти до пятнадцати (включительно). Можно, например, ввести такие обозначения:

0 — нуль	8 — восемь
1 — один	9 — девять
2 — два	$\bar{0}$— десять
3 — три	$\bar{1}$— одиннадццать
4— четыре	$\bar{2}$ — двенадцать
5 —пять	$\bar{3}$— тринадцать
6 —шесть	$\bar{4}$ — четырнадцать
7 — семь	$\bar{5}$— пятнадцать

Число сто девяносто пять, запись которого мы уже приводили в десятичной, восьмеричной и двоичной системах счисления, в шестнадцатиричной системе будет представлено

$\overline{23} = \bar{2}\cdot10^1 + 3\cdot10^0$ (здесь 10 означает шестнадцать).

Для решения задач на электронных цифровых машинах используются, в основном, следующие системы счисления: десятичная, двоичная, восьмеричная и шестнадцатиричная.

Десятичная система счисления в обычном, общепринятом виде используется в процессе подготовки задач, а также для выдачи результатов решения.

Кроме того, десятичная система счисления в специальном двоично-кодированном виде используется непосредственно в некоторых машинах для вычислений, о чем подробнее будет сказано ниже.

Двоичная система счисления применяется для представления чисел и выполнения операций в большинстве современных машин.

Основным достоинством двоичной системы счисления является возможность использования для представления каждого разряда числа любого элемента (физического прибора), имеющего только два различных устойчивых состояния, например, реле, электронных ламп и др., т. е. таких приборов, которые работают по наиболее простому и надежному принципу действия «да» или «нет» (включено или выключено).

Одно из устойчивых состояний элемента (например «да», включено) означает единицу, другое — нуль. Никаких иных цифр содержать разряд числа не может, так как двоичная система использует лишь две различных цифры 0 и 1.

Создание физических элементов, имеющих больше чем два различных устойчивых состояния, труднее, чем элементов с двумя устойчивыми состояниями.

Преимущество двоичной системы счисления перед другими системами заключается также в простоте операции умножения двоичных чисел. Так как множитель состоит только из нулей и единиц, то умножение множимого на каждый разряд множителя заключается либо в написании множимого без изменения на соответствующем разрядном месте, либо в переходе к следующему разряду множителя.

Использование двоичной системы счисления для представления чисел в электронных машинах позволяет кроме того при построении и анализе функциональных схем применять аппарат математической логики и обеспечивает существенное упрощение строения арифметических и запоминающих устройств по сравнению со случаями, когда используются другие системы счисления, (см. § 7). Неудобством, связанным с применением двоичной системы, является необходимость преобразования входных данных из десятичной системы в двоичную систему и обратного преобразования результатов вычислений.

Поскольку для подавляющего большинства математических задач характерен большой объем операций при относительно малом числе исходных данных и результатов решений, то этот недостаток двоичной системы счисления не является существенным.

Отметим, что существуют приборы, имеющие три резко ограниченных устойчивых состояния (например приборы, построенные на принципе намагничивания с использованием трех состояний: положительного, нулевого и отрицательного). Эти приборы являются трехпозиционными и с их помощью можно построить вычислительные машины, использующие для изображения чисел троичную систему счисления.

Такая машина при своем осуществлении потребовала бы меньшего количества элементов для изображения того же диапазона чисел, чем машина, использующая двоичную систему счисления.

Чтобы убедиться в этом, подсчитаем количество двухпозиционных и трехпозиционных элементов, нужное для представления диапазона чисел от нуля до девятьсот девяносто девяти.

В двоичной системе счисления число девятьсот девяносто девять записывается в виде десятиразрядного числа 1111100111. Следовательно, с помощью десяти двухпозиционных элементов можно представить любое число из указанного выше диапазона. В троичной системе счисления девятьсот девяносто девять записывается в виде семиразрядного числа 1101000. Следовательно, для представления любого числа из нашего диапазона требуется лишь семь трехпозиционных элементов.

Однако, троичная система счисления не получила широкого распространения в автоматических машинах в связи с трудностями создания достаточно быстродействующих запоминающих устройств большой емкости, построенных с использованием трехпозиционных элементов. Например, магнитные запоминающие устройства, работающие по принципу перемагничивания (с использованием двух различных состояний: положительного и отрицательного), оказались гораздо проще и надежнее, чем работающие с использованием трех состояний намагничивания.

В настоящее время может быть получена скорость избирательного перемагничивания магнитной записи двоичных цифр до 100000 и более перемен полярностей в секунду. В то же время избирательное размагничивание до нулевого уровня, требуемое для троичной системы счисления, в настоящее время не может производиться с такими скоростями. Таким образом, троичная система счисления не получила применения в вычислительных машинах общего назначения, но она имеет значение для использования в некоторых специальных случаях, например, когда требуется иметь большое запоминающее устройство для хранения редко изменяющихся данных.

Восьмеричная система счисления применяется при программировании задач для записи порядковых номеров команд, кодов операций, адресов в командах и условных чисел. (см. § 11).

Восьмеричная система счисления удобна тем, что при ее использовании достаточно просто можно осуществлять преобразование чисел в двоичную систему счисления, а также обратное преобразование из двоичной системы счисления в восьмеричную. Преобразование каждого разряда восьмеричного числа в трехзначное двоичное число дает сразу преобразование всего восьмеричного числа в двоичное число. Следует только помнить, что каждый восьмеричный разряд должен обязательно быть представлен тремя двоичными разрядами. Например, восьмеричное число 67 в двоичной системе счисления будет иметь вид 110111, а число 14—001100.

Простота этих преобразований обусловлена тем, что основание системы счисления (число восемь) представляет собой целую степень двойки. Аналогичным образом используется и шестнадцатиричная система счисления для записи программ и вспомогательных чисел.

Формы представления чисел

В цифровых машинах, в зависимости от их конструкции, могут применяться две формы представления чисел:

1. Естественная форма «с фиксированной запятой».
2. Полулогарифмическая или нормальная форма «с плавающей запятой».

В машинах с естественной формой представления чисел с фиксированной запятой числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть числа от дробной. Поясним это примерами. Будем считать, что в машине могут быть представлены шесть разрядов десятичного числа (хотя, в действительности, в большинстве машин числа представляются не в десятичной, а в двоичной системе счисления). Числа в приводимых примерах представлены в естественной форме с запятой, фиксированной после третьего цифрового разряда:

а) + 684,721	б) + 068,472	в) 4-006,847
г) + 000,685	д) + 000,068	е) + 000,007
ж) + 000,000	з) —027,022 и т. п.	

«Рассматривая эти примеры, мы видим, что чем меньше абсолютная/величина изображаемого числа, тем меньшим количеством значащих цифр изображается это число. При этом, как известно, относительная погрешность при изображении малых чисел больше, чем при изображении больших чисел.

Диапазон отличных от нуля чисел, представляемых в машинах с фиксированной запятой, сравнительно невелик (в нашем примере от 000,001 до 999,999). Всякое число, меньшее по абсолютной величине представляемых в машине (в нашем примере меньше, чем 000,001), будет записано в ней в виде нуля (так называемый машинный нуль).

Если в результате какой-нибудь операции получается число по абсолютной величине больше, чем максимальное число, которое может быть представлено в машине (в нашем примере больше, чем 999,999), то происходит явление, называемое переполнением разрядной сетки и состоящее в том, что старшие разряды этого числа машина теряет. Это обстоятельство приходится заранее учитывать при подготовке задач для решения на машине, имеющей фиксированное положение запятой, умножая входящие в задачу величины на соответствующие масштабные коэффициенты. Подбор масштабных коэффициентов представляет иногда значительные трудности.

Обычно в машинах, с фиксированным положением запятой, запятая фиксируется перед старшим разрядом числа, при этом все участвующие величины должны быть меньше единицы. Такое положение запятой удобно тем, что при действиях умножения не возникает переполнения, а также облегчается подбор масштабных коэффициентов при программировании задач.

В машинах с полулогарифмической или нормальной формой представления чисел (с плавающей запятой) каждое число изображается в виде двух групп цифр. Первая группа цифр называется мантиссой, а вторая — порядком числа. Поясним это снова на примере десятичных чисел (хотя в машинах обычно представляются числа в двоичной системе), считая, что для изображения мантиссы предусмотрено шесть разрядов, а для представления порядка — два разряда.

Чтобы число 684,720 представить в нормальной форме, его необходимо преобразовать следующим образом:

$$+ 684{,}720 = + 0{,}684720 \cdot 10^{+3}.$$

В машине это число будет записано так: +684720+03..

Группа цифр +684720 является мантиссой, а группа цифр +03 — порядком нашего числа.

Как видно из этого примера, мантисса представляет собой правильную дробь с запятой, фиксированной перед первым разрядом. Порядок является целым числом.

Легко сообразить, что изображение числа в нормальной форме не является однозначным. Например, справедливы также записи: $+684{,}720=0{,}0684720 \cdot 10^4=0{,}000684720 \cdot 10^6$ и т. п.

Этим записям отвечают следующие записи в машине (с учетом того, что для мантиссы отведено всего шесть разрядов):

+ 068472 + 04
+ 000685 + 06 и т. п.

Если в первом разряде мантиссы стоит цифра, отличная от нуля, то нормальное число называют нормализованным. Если же цифра мантиссы, стоящая в ее первом разряде, является нулем, то нормальное число называют ненормализованным.

Из сказанного выше вытекает, что нормальной формой представления числа N является его представление в виде произведения:

$$N = d \cdot 10^P,$$

где d — мантисса (правильная дробь),
p — порядок (целое число).

Число N называется нормализованным, если

$$\frac{1}{10} \le d < 1$$

Эти формулы справедливы при изображении чисел в любой системе счисления, так как основание системы счисления всегда записывается двумя знаками «10».

Можно сказать, что мантисса представляет собой последовательность цифр, изображающих число, а порядок указывает в каком месте — между какими разрядами мантиссы — надлежит поставить запятую. Поэтому машины с нормальной формой представления чисел обычно и называют машинами с «плавающей» запятой (при разных порядках числа положение запятой различно).

Нормализация чисел в машине может производиться как автоматически при вводе чисел в машину или при выполнении арифметических действий, так и по специальным командам.

При нормализации сдвигается вправо или влево запятая в мантиссе и соответствуюшим образом корректируется порядок числа.

Применение нормальной формы представления чисел (с «плавающей» запятой) позволяет получить достаточно широкий диапазон представления чисел в машине без применения масштабных коэффициентов и примерно одинаковую относительную точность представления чисел независимо от их абсолютной величины.

Недостатком нормальной формы представления чисел является увеличение количества элементов оборудования, потребных для представления чисел в машине, так как при этом помимо элементов, необходимых для представления цифровой части числа — мантиссы, требуются еще элементы для представления порядка.

Кроме того, при представлении числа в нормальной форме существенно усложняется выполнение операций сложения и вычитания по сравнению с тем случаем, когда числа представлены в естественной форме.

5. ОПЕРАЦИИ С ДВОИЧНЫМИ ЧИСЛАМИ

Как уже говорилось, в машинах обычно применяют запись чисел в двоичной системе счисления. При этом для изображения знаков чисел предусматривается один дополнительный двоичный разряд, который располагается перед разрядами самого числа.

Для чисел в нормальной форме предусматривается еще один специальный двоичный разряд для изображения знака порядка, который располагается перед группой разрядов, отведенных для представления порядка.

Положительный знак изображается нулем, помещенным в знаковый разряд, а отрицательный — единицей.

Это правило удобно тем, что знаки произведения и частного получаются путем суммирования двоичных цифр, изображающих знаки сомножителей или делимого и делителя, на одноразрядном сумматоре. Суммирование производится по правилу: 0+0=0, 0+1 = 1 + 0=1, 1 + 1=0. В последнем случае единица разряда целых двоек пропадает, так как в одноразрядном сумматоре имеется всего один разряд

Для машин с фиксированной запятой знаковым разрядом обычно является разряд целых единиц (так как в машинах с фиксированной запятой запятая обычно фиксируется перед старшим разрядом числа, и все участвующие числа по абсолютной величине меньше единицы).

Поясним сказанное примерами. Число «$+\frac{9}{64}$» в машине с фиксированной запятой, использующей двоичную систему счисления, будет записано в виде:

$$\overset{\substack{\text{Знак мантиссы}\\ \downarrow}}{0}\,\underbrace{001001}_{\text{Число}}$$

Здесь первый (знаковый) разряд содержит цифру 0, означающую знак плюс. В остальных разрядах записано само число, запятая фиксирована после знакового разряда. Таким образом эта запись обозначает + 0,001001$\left(+\frac{9}{64}\right)$.

Число „$-\frac{9}{64}$“ было бы записано в виде:

$$\overset{\substack{\text{Знак}\\ \downarrow}}{1}\,\underbrace{001001}_{\text{Число}}$$

Единица, стоящая в знаковом разряде, означает знак минус. Запись означает число—0,001001$\left(-\frac{9}{64}\right)$

В случае машины с плавающей запятой, использующей двоичную систему счисления и имеющей пять разрядов для мантиссы и три разряда для представления порядка, число „$+\frac{9}{64}$“ оказалось бы записанным в виде:

$$\underbrace{\overset{\substack{\text{Знак мантиссы}\\ \downarrow}}{0}\,1001}_{\text{Мантисса}}\quad \underbrace{\overset{\substack{\text{Знак порядка}\\ \downarrow}}{1}\,10}_{\text{Порядок}}$$

а число „$-\frac{9}{64}$“ представилось бы так:

$$\underbrace{\overset{\substack{\text{Знак мантиссы}\\ \downarrow}}{1}\,1001}_{\text{Мантисса}}\quad \underbrace{\overset{\substack{\text{Знак порядка}\\ \downarrow}}{1}\,10}_{\text{Порядок}}$$

Из приведенных примеров видно, что правила записи чисел в машине с фиксированной запятой (запятая фиксирована перед первым разрядом) и запись мантиссы в машине с плавающей запятой одинаковы.

Изображение отрицательных чисел

Необходимость выполнения действий не только с положительными, но и с отрицательными числами привела к применению трех способов кодирования чисел в автоматических машинах: прямым, дополнительным и обратным кодами. В коде числа записываются в своем обычном виде, а знаки чисел изображаются согласно указанному выше правилу (нулем и единицей).

Для положительных чисел изображение во всех трех кодах совпадает, т. е. положительные числа в дополнительном и обратном коде имеют такое же изображение, как и в прямом коде. Различие в кодах проявляется при изображении отрицательных чисел. Введение дополнительного и обратного кодов вызвано стремлением свести операцию прямого вычитания, которую трудно непосредственно осуществить в автоматических машинах, к сложению этих специально подобранных кодов.

Будем рассматривать числа в двоичной системе счисления, являющиеся правильными дробями с запятой, стоящей перед первым разрядом.

Рассмотрим подробнее прямой код.

Прямой код числа x будем обозначать символом $[\boldsymbol{x}]_{пр}$.

Если $x= +0, x_1, x_2, x_3,... x_n$ (где x_i — двоичная цифра, равная нулю или единице), то

$$[x]_{пр}=0, x_1, x_2, x_3,... x_n=x.$$

Если же $x = — 0, x_1, x_2, x_3,... x_n$, то

$$[x]_{пр}=1, x_1, x_2, x_3, ... x_n = 1 — (— 0, x_1, x_2,...x_n)=1— x.$$

Так как $0= + 0 = — 0$, то

$$[0]_{пр} = 0,0000...0= 1,000...0,$$

т. е. нуль имеет два представления в прямом коде. Формулы для $x>0$, $x = 0$ и $x<0$ можно объединить следующим образом:

$$[x]_{пр} = \begin{cases} x, \text{ если } x \geqslant 0; \\ 1 - x, \text{ если } x \leqslant 0. \end{cases} \qquad \text{(II.1)}$$

Перейдем к рассмотрению дополнительного кода. Дополнительный код числа x будем обозначать символом $[x]_{доп}$. Тогда при $x>0$ $[x]_{доп} = x$.

Чтобы получить дополнительный код для отрицательного числа, нужно в знаковом разряде поставить единицу, во всех цифровых разрядах заменить единицы нулями, а нули — единицами, затем: к младшему разряду прибавить единицу. В случае возникновения переноса из первого после запятой разряда в знаковый разряд со знаковым разрядом нужно поступить как с разрядом целых единиц.

Например,

$$[— 0,1101]_{доп} = 1,0010 + 0,0001 = 1,0011,$$

$$[— 0,1100]_{доп} = 1,0011 + 0,0001 = 1,0100.$$

Таким образом, если $x =— 0, x_1, x_2... x_n$ то, полагая $x'_i=1$, при $x_i = 0$ и $x'_i = 0$ при $x_i=1$, можем написать

$$[x]_{доп}=1, x'_1, x'_2... x'_n + \underbrace{0,00...01}_{n \text{ разрядов}}.$$

Отсюда следует

$$[x]_{доп}—x= 1, x'_1, x'_2... x'_n + 0,00...01—(—0, x_1, x_2... x_n) = 1,11...1 + 0,00...01 = 10$$

и

$$[x]_{доп}=10 +x,$$

где „10" означает число два в двоичной системе счисления. При этом по нашему правилу:

$$[0]_{доп}= + [0,00...0]_{доп} = 0,$$

$$[—0]_{доп}=[—0,00...0]_{доп} = 10—0 = 10.$$

Учитывая, что в разрядной сетке машины нет разрядов левее знакового (на который мы теперь смотрим как на разряд целых единиц), легко сообразить, что первая цифра числа 10 машиной будет утеряна, а в знаковом разряде будет стоять 0. Таким образом в машине получится $[0]_{доп} = [—0]_{доп}=0$. Объединяя вместе формулы для $x>0$, $x = 0$ и $x<0$, мы можем написать

$$[x]_{доп} = \begin{cases} x \text{ при } x \geqslant 0; \\ 10 + x \text{ при } x < 0. \end{cases} \qquad \text{(II.2)}$$

Решим теперь вопрос, как нужно складывать дополнительные коды чисел для того, чтобы получить дополнительный код их суммы. При этом будем считать, что получаемая сумма чисел по абсолютной величине меньше единицы (т. е. может быть записана без переполнения разрядной сетки).

Пусть x и y—два числа.

Очевидно, если $x \geq 0$, $y \geq 0$ (а следовательно, и $x+y \geq 0$), то формула (II.2) дает

$$[x]_{доп} + [y]_{доп} = x + y = [x + y]_{доп}.$$

Если одно из чисел x, y положительно, а другое отрицательно, то возможны два случая:

а) $x+y \geq 0$.

Тогда

$$[x]_{доп} + [y]_{доп} = 10+x+y = 10+ [x+y]_{доп}$$

Правая часть последней формулы больше, чем 10 (два), так как $x+ y>0$, но меньше, чем 11,000 (три), так как $x+ y <1$, т. е. правая часть представляет собой число вида 10,… Очевидно, что от правой части нужно отбросить 10 (два), после чего она будет представлять собой дополнительный код суммы. Однако это сводится к отбрасыванию единицы переноса, получающейся в знаковом разряде, если сложение дополнительных кодов производить как сложение обычных чисел, считая знаковые разряды разрядами целых единиц.

Итак, для того чтобы при сложении дополнительных кодов двух чисел получался дополнительный код их суммы, нужно эти коды складывать, как обычные числа, поступая со знаковыми разрядами, как с обычными разрядами целых единиц и отбрасывая единицу переноса из знаковых разрядов (если она возникает). Мы убедимся сейчас, что это правило справедливо и в остальных возможных случаях сложения дополнительных кодов.

б) $x + y<0$.

Согласно (II.2) имеем:

$$[x]_{доп} + [y]_{доп} = 10 + x + y = [x+ y]_{доп}.$$

В этом случае единица переноса из знаковых разрядов не возникает и ее отбрасывать не приходится.

Наконец рассмотрим случай, когда $x < 0$, $y < 0$. При этом, естественно, $x + y < 0$.

Формула (II.2) нам дает

$$[x]_{доп} + [y]_{доп} = 100 + x + y$$

Правая часть в последней формуле больше трех, но меньше четырех (так как $-1< x +y <0$, а 100 означает число четыре.), т. е. она имеет вид числа 11. Отбрасывая единицу переноса, возникающую из знакового разряда, мы превратим ее в число вида 1, ..., т. е. в число:

$$10+x+y = [x+y]_{доп}.$$

дополнительному коду их суммы. Сложение по описанному правилу в машине осуществляется с помощью прибора, называемого сумматором. Сумматор устроен так, что складывает дополнительные коды как обычные числа. Знаковые разряды складываются как обычные числовые разряды, а единица переноса из знаковых разрядов в этом сумматоре теряется. Приведем примеры сложения дополнительных кодов.

1) $x=0{,}1101$ $\quad [x]_{доп}=0{,}1101 \quad +\ 0{,}1101$
$\quad y=0{,}0001$ $\quad [y]_{доп}=0{,}0001 \quad\ \ \ 0{,}0001$

$x+y=0{,}1110 \quad [x+y]_{доп}=0{,}1110 \leftarrow 0{,}1110$

2) $x=+0{,}1101$ $\quad [x]_{доп}=0{,}1101 \quad +\ 0{,}1101$
$\quad y=-0{,}0001$ $\quad [y]_{доп}=1{,}1111 \quad\ \ \ 1{,}1111$

$\qquad\qquad\qquad\qquad\qquad\qquad 10{,}1100$

$x+y= \ 0{,}1100 \quad [x+y]_{доп}=0{,}1100 \leftarrow 0{,}1100$

3) $x=-0{,}1101$ $\quad [x]_{доп}=1{,}0011 \quad +\ 1{,}0011$
$\quad y=+0{,}0001$ $\quad [y]_{доп}=0{,}0001 \quad\ \ \ 0{,}0001$

$x+y=-0{,}1100 \quad [x+y]_{доп}=1{,}0100 \leftarrow 1{,}0100$

4) $x=-0{,}1101$ $\quad [x]_{доп}=1{,}0011 \quad +\ 1{,}0011$
$\quad y=-0{,}0001$ $\quad [y]_{доп}=1{,}1111 \quad\ \ \ 1{,}1111$

$\qquad\qquad\qquad\qquad\qquad\qquad 11{,}0010$

$x+y=-0{,}1110 \quad [x+y]_{доп}=1{,}0010 \leftarrow 1{,}0010$

Рассмотрим обратный код.

Обратный код числа x обозначим символом $[x]_{обр}$. Как же говорилось, обратный код положительного числа совпадает с его прямым кодом. Поэтому при $x>0$

$$[x]_{обр}= x$$

Для отрицательного числа обратный код получаем, вписывая в знаковый разряд единицу, а в цифровых разрядах, заменяя единицы нулями и нули единицами, т. е. если $x=-0, x_1, x_2,\dots x_n$, то,

полагая $x'_i = 0$ при $x_i= 1$ и $x'_i= 1$ при $x_i = 0$, получим

$$[x]_{обр}= 1,$$

Отсюда следует

$$[x]_{обр} - 1= 1, x'_1, x'_2,\dots x'_n + 0, x_1, x_2,\dots x_n = 1,\underbrace{11\dots1}_{n\ \text{знаков}}$$

Таким образом, при $x<0$ получаем

$$[x]_{\text{обр}}=10-1\cdot 10^{-n}+x \text{ („10“ -два).}$$

Если $x = 0$, то по нашему правилу имеем

$$[+0]_{\text{обр}} = [+0,00...0]_{\text{обр}} = 0,00...0,$$

а с другой стороны

$$[-0]_{\text{обр}} = [-0,00...0]_{\text{обр}} =10-1\cdot 10^{-n}+0 = 1,11...1$$

Объединяя в одну запись формулы для x>0, $x = 0$, $x<0$, можем написать

$$[x]_{\text{обр}}=\begin{cases} x, & \text{если } x\geqslant 0; \\ 10-1\cdot 10^{-n}+x, & \text{если } x\leqslant 0. \end{cases} \qquad \text{(II.3)}$$

Существует такой способ сложения обратных кодов, который приводит к обратному коду суммы. Этот способ состоит в следующем.

Если обратные коды складывать, как обычные числа, поступая со знаковыми разрядами, как с разрядами целых единиц, и если единицу переноса из знакового разряда (в случае ее возникновения) прибавлять к младшему разряду суммы кодов (это так называемый циклический перенос), то получится обратный код суммы. При этом предположено, что сумма чисел не превосходит по абсолютной величине единицы (т. е. при записи не переполняет разрядную сетку).

Действительно, если $x>0$, $y>0$, $0 < x + y < 1$, то по формуле (II.3)

$$[x]_{\text{обр}} + [y]_{\text{обр}}= x+y=[x+y]_{\text{обр}}.$$

Если одно из чисел x, y положительно, а другое отрицательно, то возможны два случая:

1. $x + y>0$. В этом случае формула (II.3) дает

$$[x]_{\text{обр}}+[y]_{\text{обр}} = 10-1\cdot 10^{-n}+x+y=10 + (x+y)-1\cdot 10^{-n}.$$

Очевидно нужно отбросить 10 и прибавить $1\cdot 10^{-n}$, чтобы последнее число приняло вид

$$x+y=[x+y]_{\text{обр}},$$

т. е. нужно осуществить циклический перенос из знакового разряда в младший разряд суммы.

2. $-1<x+y<0$. В этом случае согласно (II.3) имеем

$$[x]_{\text{обр}} + [y]_{\text{обр}}=10-1\cdot 10^{-n}+x+y = [x+y]_{\text{обр}}.$$

Здесь циклический перенос отсутствует, так как в знаковом разряде единица переноса не возникает.

Наконец, если $x<0$, $y < 0$, а значит и $x+y<0$, то

$$[x]_{\text{обр}} + [y]_{\text{обр}}=10-1\cdot 10^{-n}+x+10-1\cdot 10^{-n}+y= 10 +[10-1\cdot 10^{-n}+(x+y)]-1\cdot 10^{-n}.$$

Правая часть полученного равенства после циклического переноса примет вид

$$10-1\cdot 10^{-n}+(x+y)-1\cdot 10^{-n}+1\cdot 10^{-n}=10-1\cdot 10^{-n}+(x+y)=[x + y]_{\text{обр}}.$$

Таким образом, справедливость указанного выше правила сложения обратных кодов установлена. Сложение по этому правилу производится прибором, называемым сумматором с круговым или циклическим переносом.

Примеры сложения обратных кодов ($n = 4$).

$$\begin{array}{llr}
1)\ x=0,1101 & [x]_{\text{обр}}=0,1101 & +\ 0,1101 \\
y=0,0001 & [y]_{\text{обр}}=0,0001 & 0,0001 \\
\hline
x+y=0,1110 & [x+y]_{\text{обр}}=0,1110 \leftarrow & 0,1110
\end{array}$$

$$\begin{array}{llr}
2)\ x=0,1101 & [x]_{\text{обр}}=0,1101 & +\ 0,1101 \\
y=-0,0001 & [y]_{\text{обр}}=1,1110 & 1,1110 \\
\hline
 & & 10,1011 \\
 & & \text{циклический перенос} \\
 & & 0,1011 \\
 & & 1 \\
\hline
x+y=0,1100 & [x+y]_{\text{обр}}=0,1100 \leftarrow & 0,1100
\end{array}$$

$$\begin{array}{llr}
3)\ x=-0,1101 & [x]_{\text{обр}}=1,0010 & +\ 1,0010 \\
y=0,0001 & [y]_{\text{обр}}=0,0001 & 0,0001 \\
\hline
x+y=-0,1100 & [x+y]_{\text{обр}}=1,0011 \leftarrow & 1,0011
\end{array}$$

$$\begin{array}{llr}
4)\ x=-0,1101 & [x]_{\text{обр}}=1,0010 & +\ 1,0010 \\
y=-0,0001 & [y]_{\text{обр}}=1,1110 & 1,1110 \\
\hline
 & & 11,0000 \\
 & & \text{циклический перенос} \\
 & & 1,0000 \\
 & & 1 \\
\hline
x+y=-0,1110 & [x+y]_{\text{обр}}=1,0001 \leftarrow & 1,0001
\end{array}$$

Вычитание числа y из числа x сводится к алгебраическому сложению этих чисел. При этом знак у второго слагаемого меняется на обратный и производится алгебраическое сложение чисел в дополнительном или обратном коде в зависимости от конструкции сумматора.

Признаки переполнения разрядной сетки

В машинах с фиксированной запятой все числа должны быть по абсолютной величине меньше единицы.

Однако при сложении двух чисел, каждое из которых меньше единицы, может получиться число, большее единицы, т. е. может произойти переполнение разрядной сетки.

Такие случаи недопустимы для нормальной работы машины с фиксированной запятой и должны сразу обнаруживаться с тем, чтобы машина могла быть остановлена для введения необходимых масштабных множителей, которые бы устранили возможность переполнения.

Рассмотрим два способа определения наличия переполнения,

В первом способе для изображения знаков чисел применяется один разряд. При этом, если $0<x<1$, $0<y<1$ и $x+y>1$, то в разряде знака суммы появится 1.

Пример:

$$x=+0{,}110110;\quad y=+0{,}010110.$$

$$\begin{array}{r} + \begin{array}{r} 0{,}110110 \\ 0{,}010110 \end{array} \\ \hline 1{,}001100 \end{array}$$

Если же $—1<x<0$, $—1<y<0$, $|x+y|>1$, то наоборот, признаком переполнения будет служить появление в знаковом разряде суммы цифры 0.

Пример:

$$x=-0{,}110110,\qquad y=-0{,}010110,$$
$$[x]_{доп}=1{,}001010,\qquad [y]_{доп}=1{,}101010$$

$$\begin{array}{r} + \begin{array}{r} 1{,}001010 \\ 1{,}101010 \end{array} \\ \hline 10{,}110100 \end{array} \rightarrow 0{,}110100.$$

так как старший разряд теряется.

Можно показать справедливость этого правила в общем случае. Например, для дополнительного кода при $—1<x<0$ и $-1<y<0$

$$[x]_{доп}=10+x,\qquad [y]_{доп}=10+y,$$
$$[x]_{доп}+[y]_{доп}=100+(x+y).$$

Если $—10<x+y<—1$, то

$$10<[x]_{доп}+[y]_{доп}<11.$$

Таким образом, всегда будет получаться число больше двух, но меньше трех, т. е. число вида 10, и поэтому в разряде знака суммы будет получаться цифра 0.

Этот способ определения переполнения неудобен, так как требует запоминания знаков слагаемых и сравнения их со знаком суммы.

Во втором способе обнаружения переполнения применяется так называемый модифицированный код, при котором знаки чисел изображаются двумя двоичными разрядами: например, число +0,110110 будет иметь вид 00,110110; число —0,110110 будет иметь в прямом коде вид 11,110110.

Отрицательные числа могут быть представлены модифицированным дополнительным кодом

$$[x]_{доп.\,мод}=100+x$$

или модифицированным обратным кодом

$$[x]_{доп.\,мод}=100-1.10^{-n}+x$$

в зависимости от конструкции применяемого сумматора.

Сложение чисел в модифицированном дополнительном коде осуществляется так же, как и в обычном дополнительном коде с той лишь разницей, что отбрасывается единица переноса, возникающая из старшего знакового разряда.

Сложение чисел в модифицированном обратном коде осуществляется так же, как и в обычном обратном коде с той лишь разницей, что циклический перенос осуществляется из старшего знакового разряда в младший цифровой разряд.

При использовании модифицированных кодов (обратного и дополнительного) для всякого числа меньшего по абсолютному значению, чем единица, в знаковых разрядах будут либо два нуля (00,...), либо две единицы (11,....).

Появление в знаковых разрядах при сложении чисел сочетаний разных цифр 10 или 01 будет являться признаком переполнения.

Пример: Сложение в модифицированном дополнительном коде.

1) $x=+0{,}110110,$ $y=+0{,}010110$

$$\begin{array}{r} + \begin{array}{r} 00{,}110110 \\ 00{,}010110 \end{array} \\ \hline 01{,}001100 \end{array}$$

2) $x=-0{,}111011,$ $y=-0{,}010110$

$$\begin{array}{r} + \begin{array}{r} 11{,}000101 \\ 11{,}101010 \end{array} \\ \hline 10{,}101111 \end{array}$$

Сложение и вычитание чисел в нормальной форме (с плавающей запятой)

Для сложения чисел, заданных в нормальной форме, необходимо сначала выравнить порядки слагаемых, а затем произвести сложение мантисс и нормализацию результата.

Выравнивание порядков заключается в том, что порядок меньшего числа увеличивается на разность порядков, а мантисса меньшего числа сдвигается вправо на число разрядов, равное разности порядков. Сложение мантисс производится по правилам сложения двоичных чисел в машинах с фиксированной запятой. Сумма мантисс даст мантиссу суммы, а порядок суммы принимается равным порядку большего числа.

В результате • сложения мантисс может оказаться, что сумма мантисс не является нормализованным числом. В этом случае производится сдвиг мантиссы в нужную сторону и корректировка порядка суммы. При этом также удобно пользоваться модифицированным дополнительным или обратным кодом, так как признак переполнения разрядной сетки совпадает с признаком нарушения нормализации влево (т. е. получения мантиссы суммы большей по абсолютной величине, чем 1.)

Признаком нарушения нормализации вправо (т. е. получения мантиссы суммы меньшей по абсолютной величине, чем 1/2) является наличие двух одинаковых цифр 0,0 или 1,1 в двух соседних разрядах: младшем знаковом разряде и в старшем цифровом разряде. Этот признак справедлив для обоих кодов: дополнительного и обратного.

Следует заметить, что при выполнении нормализации числа после сдвига вправо в старший знаковый разряд должна быть занесена та цифра, которая находилась в этом разряде до сдвига.

Пример сложения мантисс в модифицированном дополнительном коде:

$x = +\,0{,}110110 \cdot 10^{10}$; $\qquad$ $y = -\,0{,}101011 \cdot 10^{10}$ („10“—два)

$$\begin{array}{r} +\ 00{,}110110 \\ 11{,}010101 \\ \hline 00{,}001011 \end{array}$$

Получение в последнем примере двух одинаковых цифр 0,0 в младшем знаковом и старшем цифровом разряде свидетельствует о нарушении нормализации вправо, сумма будет равна (после сдвига мантиссы на два разряда влево) $00{,}101100 \cdot 10^{0} = +0{,}101100$.

Следует заметить, что в машинах с плавающей запятой предусматривается округление. В тех случаях, когда происходит выход мантиссы за допустимое число разрядов со стороны младших разрядов, лишние разряды отбрасываются и, если старший из отбрасываемых разрядов равен единице, к младшему остающемуся разряду прибавляется единица.

Пример сложения мантисс в дополнительном коде со сдвигом и округлением.

$x = 0{,}101110 \cdot 10^{10}$; $\qquad$ $y = 0{,}100101 \cdot 10^{10}$

$$\begin{array}{r} +\,11{,}010010 \\ 11{,}011011 \\ \hline 10{,}101101 \end{array}$$

Получение разных цифр в знаковых разрядах (10) свидетельствует о нарушении нормализации влево.

Сумма будет равна (после сдвига на один разряд вправо и сохранения в старшем знаковом разряде единицы) $=11{,}0101101 \cdot 10^{11} = = -0{,}1010011 \cdot 10^{11}$. После округления (отбрасываем единицу седьмого после запятой разряда и прибавляем единицу в шестой разряд) получим: $-0{,}101010 \cdot 10^{11}$.

Вычитание чисел в нормальной форме так же, как и в машинах с фиксированной запятой, сводится к алгебраическому сложению их обратных или их дополнительных кодов.

Умножение чисел с фиксированной запятой

Умножение двоичных чисел в машинах обычно осуществляется в прямом коде.

Знак произведения получается путем суммирования знаков сомножителей на одноразрядном двоичном сумматоре по правилу:

$0+0=0, \quad 1+0=1; \quad 0+1=1; \quad 1+1=10=0.$

Так как в сумматоре имеется всего один разряд, то в последнем случае в результате сложения получается нуль, т. е. знак плюс.

Умножение цифровых частей чисел в двоичной системе счисления осуществляется весьма просто и заключается в том, что множимое последовательно сдвигается влево на количество разрядов, равное номерам значащих разрядов множителя, и полученные таким образом частные произведения суммируются.

Пример:

$$\begin{array}{r} \times\ 0{,}110110 \\ 0{,}101011 \\ \hline 110110 \\ 110110 \\ 110110 \\ 110110 \\ \hline 0{,}100100010010 \end{array}$$

Умножение чисел в нормальной форме

Умножение чисел, заданных в нормальной форме, осуществляется в три приема.

1. Сложение цифр, изображающих знаки сомножителей, в результате чего получается знак произведения.
2. Алгебраическое сложение порядков сомножителей, которое дает порядок произведения.
3. Умножение мантисс сомножителей, в результате чего получается мантисса произведения.

Умножение мантисс ничем не отличается от умножения чисел в машинах с фиксированной запятой.

Если в результате перемножения мантисс сомножителей получится ненормализованное число, то производится нормализация, т. е. сдвиг мантиссы и соответствующее изменение порядка произведения.

Деление чисел в машинах с фиксированной запятой

Рассмотрим случай деления чисел в прямом коде и положим, что запятая расположена перед первой цифрой. В этом случае делимое и делитель должны быть меньше единицы и делимое меньше делителя, чтобы частное получилось также меньше единицы.

Определение знака частного производится так же, как и при умножении, путем сложения знаков делимого и делителя на одноразрядном сумматоре. Деление в двоичной системе осуществляется таким же образом, как и в десятичной системе. Цифры частного определяются последовательно, начиная со старшего разряда, путем вычитания делителя из остатка, полученного от предыдущего вычитания и сдвинутого на один разряд влево.

При определении первой цифры частного «остатком» является все делимое, которое перед вычитанием делителя сдвигается на один разряд влево (точнее делитель при каждом вычитании сдвигает на один разряд вправо).

Каждая цифра частного может принимать только два значения: О или 1. Если при вычитании делителя из k–го остатка разность будет положительна, то (k+1)–ая цифра частного после запятой будет 1. Если разность получилась отрицательная, то на месте (k+1)–ой цифры в частном ставится нуль, а k–ый остаток сдвигается еще на один разряд влево и из него вычитается делитель. Таким путем определяется (k+2)–ая цифра частного.

Этот процесс повторяется до тех пор, пока в частном не получится нужное количество цифр. Остаток от деления чисел по такому способу получается сдвинутым на *n* разрядов влево, где *n* — количество цифр в частном.

Рассмотрим пример:

```
0,1010010      |0,101101
  101101       |--------
 -------        0,11101
 1001010
  101101
 -------
  0111010
   101101
  -------
   0011010
    101101
   -------
  — 010011
   --------
      110100
      101101
      ------
      000111
```

При определении четвертой цифры частного остаток получился отрицательный, поэтому ставим на место четвертой цифры в частном нуль, возвращаемся к третьему остатку, сдвигаем его еще на один разряд влево и вычитаем из него делитель для определения пятой цифры частного.

Таким образом получим частное 0,11101 и остаток 0,000111 • 10~^{101}. Следует заметить, что при выполнении деления на машинах последовательные вычитания делителя осуществляются путем сложения частного с обратным или дополнительным кодом делителя, а возвращение к предыдущему остатку, после получения отрицательного остатка, происходит путем прибавления делителя к отрицательному остатку.

Деление чисел, заданных в нормальной форме

Деление чисел в нормальной форме осуществляется в три приема.

1. Определение знака частного путем сложения знаков делимого и делителя на одноразрядном сумматоре.
2. Вычитание порядка делителя из порядка делимого с учетом знаков порядков. В результате получается порядок частного.
3. Деление мантиссы делимого на мантиссу делителя, дающее мантиссу частного.

Деление происходит таким же образом, как и деление чисел I в машинах с запятой, фиксированной перед первой цифрой, однако при этом не требуется, чтобы делимое было меньше делителя.

В результате деления может получиться ненормализованное число, большее единицы или меньшее половины. Ненормализованная мантисса частного автоматически сдвигается в нужную сторону и соответственно производится корректировка порядка частного.

6. ДВОИЧНО-КОДИРОВАННЫЕ ДЕСЯТИЧНЫЕ СИСТЕМЫ

Наряду с двоичной системой счисления в быстродействующих цифровых машинах широкое применение получили и так называемые двоично-кодированные десятичные системы.

В этих системах основанием является число десять, но каждая из десяти десятичных цифр (0, 1 ,2 ... 9) изображается при помощи двоичных цифр, т. е. каждая десятичная цифра кодируется двоичными цифрами. При этом для кодирования десяти десятичных цифр § должны быть выбраны десять каких-нибудь различных комбинаций двоичных цифр 0 и 1.

Так как наименьшее количество двоичных цифр, которое позволяет получить не меньше чем десять различных двоичных чисел, равно четырем, то для представления каждой десятичной цифры в двоично-кодированных системах требуется не менее, чем четыре двоичных цифры.

Таким образом, четверками двоичных цифр (их называют еще тетрадами) могут быть изображены все десять десятичных цифр.

Для того чтобы определить тот или иной способ двоичного кодирования, выбирается определенное правило, которое ставит каждой десятичной цифре в однозначное соответствие определенную тетраду. Так как всего различных тетрад может быть 16, а используется из них для представления десятичных цифр только 10, то некоторые из тетрад оказываются лишними, т. е. возможности четырехзначного двоичного кодирования используются в десятичной системе счисления не полностью.

По сравнению с чисто двоичной системой счисления двоично-кодированные десятичные системы оказываются примерно на 20% менее экономичными в смысле потребного количества двухпозиционных элементов.

Для представления n-разрядного десятичного числа в двоичной кодированной системе требуется $p = 4n$ двоичных разрядов. В то же время для представления n-разрядного десятичного числа в двоичной системе требуется приблизительно $q = \frac{n}{\lg 2}$ разрядов. Отсюда получим: $p \approx 4q\lg 2 = 1{,}204\,q$, т. е. количество разрядов для представления n-разрядного десятичного числа в двоично-кодированной системе счисления примерно на 20% больше требуемого количества разрядов в двоичной системе.

Двоично-кодированные системы счисления применяются как для представления чисел и выполнения операций непосредственно автоматических машинах, так и в качестве промежуточных систем счисления в машинах, работающих по двоичной системе счисления, при переходе от десятичной системы к двоичной и обратно.

Применение двоично-кодированных систем счисления позволяет при построении вычислительных и запоминающих устройств машин сохранить преимущества двоичной системы счисления в части использования физических приборов, имеющих существенно двоичный характер, и в то же время обеспечивает удобства десятичной системы счисления, значительно упрощая перевод чисел из одной системы в другую.

Из 16 различных тетрад можно выбрать 10 тетрад, необходимых для кодирования десяти десятичных цифр, многими различными способами. (Общее число способов равно $\frac{16!}{6!} \approx 2{,}9 \cdot 10^{10}$).

Из этих способов наибольший интерес для применения в автоматических машинах представляют способы, приведенные в табл. II.1 и удовлетворяющие следующим требованиям (за исключением способа *а,* который не удовлетворяет 4-му требованию.)

Т а б л и ц а II .1

Десятичные цифры	Системы кодирования			Десятичные цифры	Системы кодирования		
	8421	2421	(8421) ± 3		8421	2421	(8421) ± 3
	а	б	*в*		*а*	*б*	*в*
0	0000	0000	0011	5	0101	1011	1000
1	0001	0001	0100	6	0110	1100	1001
2	0010	0010	0101	7	0111	1101	1010
3	0011	0011	0110	8	1000	1110	1011
4	0100	0100	0111	9	1001	1111	1100

1. Большей десятичной цифре должно соответствовать и большее двоичное число, изображаемое тетрадой, если тетраду прочесть как двоичное число.

2. Должна быть обеспечена возможность определения величины закодированной десятичной цифры непосредственно по виду соответствующей тетрады, для чего каждому разряду тетрады должен быть поставлен в соответствие определенный постоянный «вес» таким образом, чтобы суммирование произведений цифр, образующих тетраду, на их веса давало бы значение закодированной цифры.

В столбце *в* табл. II.1 приведена система тетрад, для которой это правило видоизменено. Значение закодированной цифры в этом случае определяется путем суммирования произведений двоичных цифр, образующих тетраду, на их веса и прибавлением к полученному результату определенной постоянной для данного способа кодирования поправки.

3. Должно иметь место однозначное соответствие между четностью кодируемых десятичных цифр и четностью двоичных чисел, изображаемых соответствующими тетрадами, рассматриваемыми как двоичные числа.

Иначе говоря, четным десятичным цифрам должны соответствовать четные тетрады, а нечетным десятичным цифрам — нечетные тетрады (или, наоборот, четным десятичным цифрам должны соответствовать нечетные тетрады, а нечетным десятичным цифрам — четные тетрады). Последний двоичный разряд тетрады при этом характеризует четность или нечетность закодированной десятичной цифры.

4. Для любых двух десятичных цифр, дополняющих друг друга до 9, соответствующие тетрады должны быть также дополнительными, т. е. получаться одна из другой путем замены 0 на 1 и 1 на 0.

В этом случае получение обратного кода осуществляется весьма просто путем поразрядного инвертирования *

* Инвертирование („перевертывание") означает замену нулей единицами, а единиц — нулями.

всех тетрад, изображающих число.

Из способов кодирования, представленных в табл. II.1, этому требованию не отвечает способ *а.*

Это требование является важным для машин, работающих в двоично-кодированной десятичной системе, в которых вычитание заменяется сложением с обратным кодом числа.

Следует заметить, что рассмотренный выше для случая двоичной системы счисления порядок вычитания путем сложения чисел в обратном коде на сумматоре, имеющем циклическую передачу из старшего разряда в младший, сохраняется и для других систем счисления и, в частности, для десятичной системы.

Пусть x и y—два n-разрядных числа в системе счисления с основанием B, а $[y]_{обр} = B^{n+1}—1—y$ —обратный код числа y, т. е. число, полученное поразрядным дополнением цифр числа y до B—1. При этом в крайнем слева (n+1)–ом разряде, в котором находился 0, появится цифра B—1. Этот разряд является знаковым. Тогда разность $x—y$ будет равна $x+[y]_{обр}$ и может быть получена на (n+1)–разрядном сумматоре, имеющем циклическую передачу со старшего разряда на младший.[**]

Действительно: $x + y = x + B^{n+1}—1—y$, так как единица (n+2)–го разряда (B^{n+1}) при циклической передаче превращается в единицу младшего разряда.

Если x > y, то

$$B^{n+1} —1—y > B^{n+1} —1— x$$

и

$$x+[y]_{обр} = x + B^{n+1} —1—y > B^{n+1} —1,$$

т. е. при вычитании меньшего числа из большего появится единица циклического переноса и результат будет положительный, представленный в прямом коде, что будет показано наличием нуля в крайнем слева (n+1)–ом разряде (знаковом).

Если x<_y, то

$$[x]_{обр.} > [y]_{обр.} \quad \text{и} \quad x+[y]_{обр.} < B^{n+1}—1,$$

т. е. при вычитании большего числа из меньшего единица циклического переноса отсутствует и результат будет отрицательный, представленный обратным кодом, что будет показано наличием цифры B—1 в крайнем слева (n+ 1)–ом разряде.

Также применим к любым системам счисления и способ вычитания путем сложения чисел в дополнительном коде на (n + 1)–разрядном сумматоре без циклической передачи.

Наиболее естественным и простым способом двоичного кодирования является так называемая двоично-десятичная система, представленная в столбце *а* табл. II.1. В этой системе десятичные цифры изображаются просто их эквивалентами в двоичной системе счисления. Например, число 647 будет иметь вид: 0110, 0100, 0111. Эта система кодирования имеет широкое применение в настоящее время в автоматических машинах, главным образом, как промежуточная система для перехода от десятичной системы к двоичной и для обратного перехода. Если применять эту систему непосредственно для вычислений в машинах, то для получения дополнений до 9, необходимых при выполнении вычитания, потребуются сравнительно сложные схемы преобразования, что является недостатком этой системы. Ввиду простоты и естественности способа кодирования, эта система используется в дальнейшем как основная для выражения через нее других систем кодирования. Введем следующее обозначение: если x — значение кодируемой десятичной цифры, то $p(x)$ будет представлять значения соответствующей тетрады, рассматриваемой как двоичное число.

Очень интересной является система кодирования, приведенная в столбце *б* табл. II.1. Это единственная система из приведенных в таблице, которая удовлетворяет полностью всем четырем указанным выше требованиям без каких-либо оговорок. Веса в этой системе соответственно равны: 2—4—2—1.

Достоинством этой системы является возможность простого получения дополнений до 9 путем инвертирования двоичных цифр каждой тетрады. Например, 5+4 = 9; $p(5)$ = 1011; $p(4)$=0100 (см. табл. II.1). Из табл. II.1 видно, что для системы кодирования, приведенной в столбце *б,* $p(x)=x$ для x<5 и $p\{x)=x+6$ для x>5. Например, число 639 в этой системе будет иметь вид: 1100, 0011, 1111. Система кодирования, представленная в столбце *в* табл. П.1, называется «кодом с излишком три». В этой системе кодирования веса разрядов тетрады равны соответствующим весам разрядов основной системы *(а),* а значения кодируемых десятичных цифр связаны со значениями двоичных чисел, изображаемых соответствующими тетрадами следующим образом: $p(x)=x+3$ или, если e_1 e_2, e_3, e_4—значения двоичных цифр тетрады, то $x=8e_1+4e_2+2e_3+e_4—3$.

Как видно из табл. II.1, эта система кодирования обладает той особенностью, что ни одна тетрада не может состоять только из единиц или нулей. Появление в какой-либо тетраде только нулей или единиц будет свидетельствовать о неправильности работы машины. Эта система, так же как и система б, позволяет получать дополнение до 9 путем инвертирования всех четырех разрядов соответствующей тетрады.

Выполнение операций с двоично-кодированными числами

Все арифметические действия над многозначными числами сводятся в конечном счете к поразрядному, так сказать «по-тетрадному», сложению с учетом возникающих переносов. Поэтому мы ограничимся рассмотрением этого случая.

Пусть $p(x)$ и $p(y)$—тетрады, изображающие две складываемые десятичные цифры x и y, а z — значение переноса из соседнего справа младшего разряда (этот перенос может быть равен 0 или 1). Правило сложения будет заключаться в следующем. Необходимо по значениям тетрад складываемых чисел и значению переноса определить тетраду суммы таким образом, чтобы выполнялись условия:

[**] Напомним, что в данном случае каждый разряд — это тетрада (четыре двоичных разряда).

$$p(x) + p(y)+ z = p\ (x\ \ +y+z) \text{ для } x+ y+z< 10;$$
$$p(x) +p\ (y) + z = 16 + p(x+ y + z— 10) \text{ для } x+y+z\geq 10,$$

т. е. тетрада, изображающая результат сложения, должна получаться по тому же правилу кодирования, какое принято для кодирования исходных цифр.

В случае, если значение суммы получается большим или равным 10, должен быть образован перенос в соседнюю слева тетраду, изображающую старший разряд, а в данном разряде должна быть записана тетрада десятичного числа, соответствующая разности между полученной суммой и десятью. Появление единицы переноса соответствует переполнению данной тетрады, т. е. значению числа двоичной системы счисления 10000 (шестнадцать).

В общем случае указанное правило сложения двоично-кодированных десятичных цифр не может быть

осуществлено путем простого суммирования двух исходных тетрад и единицы переноса по правилам двоичной арифметики. Необходимо к получающейся таким путем сумме прибавлять специальную поправку, зависящую от способа кодирования. Для приведенных в табл. II.1 способов кодирования значения необходимых поправок определяются следующим образом.

Двоично-десятичная система

Для двоично-десятичной системы справедливо соотношение:

$$p\ (x+y+z) =x+y+z.$$

При этом для выполнения указанного выше правила сложения (Необходимо, чтобы тетрада суммы S получалась согласно следующим правилам:

$$S = x+y+z, \qquad \text{если } x+y+z<10;$$
$$S = x+y+z+6, \qquad \text{если } x+y+z \geq 10,$$

т. е. когда сумма складываемых десятичных цифр и единицы переноса из младшего разряда больше или равна десяти, для получения соответствующей тетрады суммы и единицы переноса в старший разряд необходимо прибавлять в качестве поправки число 0110 (шесть).

Код с излишком 3

Для этого кода имеет место соотношение

$$p(x)+p(y)+z = (x + 3)+(y+3)+z = x+y+z+6.$$

Необходимо, чтобы при сложении выполнялись равенства

$$S = x+y + z+3 \qquad \text{для } x+y+z< 10;$$
$$S = x+y+z+9 \qquad \text{для } x+y+z\geq 10.$$

Отсюда видно, что необходимая поправка в случае, если $x+y+z< 10$, равна — 0011 (минус три) и в случае, если $x+y+z\geq 10$, равна + 0011 (плюс три).

Код с весами 2—4—2—1

Значения поправок при сложении легко получаются путем сравнения значения суммы, получающейся при обычном сложении исходных тетрад и единицы переноса, со значением тетрады суммы, полученной по правилам кодирования в данной системе с учетом передачи единицы переноса в старший разряд (в случае, когда сумма складываемых цифр больше или равна 10).

Рассмотрим случаи, возникающие при этом. Напомним, что в данной системе кодирования имеют место соотношения:

$$p(x) = x \quad \text{для } x<5 \quad \text{и } p(x) = x+6 \quad \text{для } x\geq 5.$$

Если:

а) $x<5$; $y<5$; $x+y+z<5$, тогда

$$p(x)+p(y) + z = x+y + z = p(x+y+z),$$

т. е. поправки в этом случае не требуется;

б) $x<5$; $y<5$; $x+y+z\geq 5$, тогда

$$p(x)+p(y) + z = x + y + z = p(x+y + z) — 6,$$

т. е. для выполнения принятого правила кодирования необходимо иметь поправку + 0110 (плюс шесть);

в) $x\geq 5$; $y<5$; $5\leq x + y+z< 15$, тогда

$$p(x)+p(y) + z = x + 6+y+z = p(z +y + z).$$

В этом случае поправки не требуется. При этом, если $x+y+z \geq 10$, то $x+y +2 + 6 \geq 16$ и необходимая единица переноса получается автоматически. Также автоматически обеспечивается и выполнение правила кодирования. Так как $x+y+z<15$, то $x+y+z+6<21$ и, если образуется единица переноса, разность $[p\ (x) +p\ (y) +z]—16$ будет меньше 5 и необходимая форма кодирования может быть представлена

$$p(x)+p(y) + z— 16 = (x+6)+y + z—16 = x+y + z—10= p(x+ y + z—10)$$

(так как $x + y + z — 10 < 5$).

Ясно, что то же самое будет иметь место и при $x<5$, $y≥5$ и $5≤x + y + z< 15$;

г) $x≥5$; $y≥5$; $10≤x + y + z< 15$, тогда

$$p(x) + p(y) + z = (x + 6) + (y + 6) + z = p(x + y + z) + 6.$$

Так как результат сложения согласно рассматриваемому правилу кодирования должен иметь вид

$$p(x) + p(y) + z=16 + p(x+y+z—10) == x + y+z + 6 = p(x+y+z),$$

[так как $x + y + z— 10 < 5$ и $p(x+y+z — 10) = x+y + z — 10$], то в этом случае необходимо иметь поправку —0110 (минус шесть);

д) $x≥5$; $y≥5$; $15≤x+y + z$, при этом справедливо равенство

$$p(x) + p(y) + z = p(x + y + z) + 6.$$

Согласно правилу кодирования результат должен иметь вид

$$p(x) + p(y) + z=16 + p(x + y + z)—10 == 16 + [(x+y+z—10) + 6] = x + y+z+12 = p(x+y+z)+6,$$

[так как $x+y+z — 10≥5$ и $p(x+y+z—10) = (x + y+z—10)+6$], т. е. в этом случае поправки не требуется.

Таким образом, поправки требуются только для случаев б и г, при этом легко заметить, что значения тетрад суммы, требующие поправок, совпадают как раз с теми значениями четырехзначных двоичных чисел, которые не используются в данной системе кодирования: 0101; 110; 0111; 1000; 1001; 1010. Это следует из того, что в случае *б* значение суммы $p(x)+p(y)+z$ лежит от 5 до 9, а в случае *г* — от 22 до 26 (т. е. от 6 до 10, если учесть уменьшение значения суммы на 16 за счет переноса).

Кроме рассмотренных систем кодирования тетрадами двоичных цифр в некоторых машинах используются системы двоичного кодирования большим числом Цифр В качестве примера можно привести так называемую двоично-пятиричную систему кодирования, приведенную в табл. П.2

Как видно из таблицы, крайний слева разряд не является необходимым для получения десяти различных комбинаций двоичных цифр.

Однако введение этого лишнего разряда позволяет выполнить условие, состоящее в том, чтобы в каждой кодирующей комбинации из 5 двоичных цифр были обязательно две цифры равны единице и три цифры равны нулю, что используется для контроля правильности работы машины.

Таблица П.2

0	10001	5	00110
1	10010	6	11000
2	00011	7	01001
3	10100	8	01010
4	00101	9	01100

Перевод чисел из десятичной системы счисления в двоичную

При вводе чисел в машину возникает необходимость переводить числа из десятичной системы счисления в двоичную. При этом в качестве промежуточной системы счисления используется двоично-десятичная система счисления.

Перевод чисел из десятичной системы счисления в двоично-десятичную осуществляется вне машины на специальных внешних устройствах. Перевод чисел из двоично–десятичной системы в двоичную осуществляется машиной по специальной стандартной подпрограмме.

Пусть задано поразрядное десятичное число

$$x_{(10)} = \sum_{n=1}^{m} a_n \cdot 10^{k-n},$$

где k — показывает положение запятой, отделяющей целую часть числа от дробной.

Для того чтобы получить двоично-десятичную запись этого числа, необходимо каждую цифру a_n десятичного числа записать в виде четырехразрядного двоичного числа, что даст коэффициент при 10^{k-n}. В машину обычно заранее вводятся и постоянно хранятся необходимые степени десяти, переведенные в двоичную систему счисления.

Перевод из двоично-десятичной системы счисления в двоичную будет заключаться в умножении последовательных четверок двоичных цифр двоично-десятичного числа на соответствующие степени Десяти и в суммировании полученных произведений.

Так как запись числа в двоично-десятичной системе приблизительно на 20% длиннее записи в двоичной системе счисления, то ввод двоично-десятичных чисел в машину может быть произведен с меньшим числом разрядов, чем ввод двоичных чисел.

Перевод чисел из двоичной системы счисления в двоично-десятичную

Рассмотрим порядок перевода чисел в машинах с фиксированной запятой. Будем считать, что запятая фиксирована перед первой цифрой числа.

Пусть задано двоичное число $x_{(2)} = 0, x_1, x_2, x_3,...x_n$, где $x_i = 0$ или 1. Нужно найти двоично-десятичную запись числа $x_{(2)}$, которая будет иметь вид $x_{(10)} = \sum_{n=1}^{m} a_n \cdot 10^{k-n}$, где a_k — тетрады, отвечающие цифрам 0,1, 2, 3, 4, 5, 6, 7, 8, 9 (10 —десять).

Очевидно, первая цифра десятичного числа получится, если заданную двоичную дробь с запятой, стоящей перед ее первой цифрой, умножить на представленное в двоичной системе число десять (получим «первое произведение») и выделить целую часть полученного результата. Эта целая часть, записанная в виде четырехразрядного двоичного числа, является первой тетрадой искомого двоично-десятичного числа. Дробную часть нашего «первого произведения» следует выделить и снова умножить на десять (получится «второе произведение») и опять выделить целую часть. Таким путем будет найдена вторая тетрада искомого двоично-десятичного числа. И так далее, до тех пор, пока не будет все двоичное число записано в двоично-десятичной системе счисления.

В машинах с плавающей запятой сначала находится значение порядка *P* десятичного числа из условия $10^{P-1} < x_{(2)} < 10^{P}$, затем число $x_{(2)}$ множится на 10^{-P} (десять в степени минус *P)* и получается значение мантиссы десятичного числа в двоичной системе счисления **с** запятой, фиксированной перед первой цифрой. Полученное двоичное число переводится в двоично-десятичную систему по приведенному выше правилу.