

СОДЕРЖАНИЕ

Введение

1. Понятие системы счисления. Классификация систем счисления.

Позиционные и непозиционные системы счисления

1.1 Непозиционные системы счисления

1.2 Позиционные системы счисления

2. Перевод чисел из одной системы счисления в другую

2.1 Перевод целых чисел из одной позиционной системы счисления в другую

2.2. Перевод правильных дробей

2.3 Перевод неправильных дробей

2.4 Перевод чисел из системы счисления в систему с кратным основанием

3. Выбор системы счисления для применения в ЭВМ

4. Двоичная система счисления

4.1 Навыки обращения с двоичными числами

5. Формы представления двоичных чисел в ЭВМ

6. Точность представления чисел в ЭВМ

Вывод

Литература

Введение

Тема реферата по курсу «Прикладная теория цифровых автоматов» - «Представление численной информации в ЭВМ. Системы счисления».

Цель написания реферата:

ознакомится с понятием системы счисления; классификацией систем счисления; переводом чисел из одной системы счисления в другую; выбором системы счисления для применения в ЭВМ; двоичной системой счисления; формами представления двоичных чисел в ЭВМ; точностью представления чисел в ЭВМ и др.

1. Понятие системы счисления. Классификация систем счисления.

Позиционные и непозиционные системы счисления

Системы счисления были созданы в процессе хозяйственной деятельности человека, когда у него появилась потребность в счете, а по мере развития научной и технической деятельности возникла также необходимость записывать числа и производить над ними вычисления

Системой счисления называется совокупность символов и приемов, позволяющих однозначно изображать числа. Или, в общем случае, это специальный язык, алфавитом которого являются символы, называемые цифрами, а синтаксисом - правила, позволяющие однозначно сформировать запись чисел. Запись числа в некоторой системе счисления называется кодом числа. В общем случае число записывается следующим образом:

$$A = a_n a_{n-1} \dots a_2 a_1 a_0$$

Отдельную позицию в записи числа принято называть разрядом, а номер позиции - номером разряда, количество разрядов в записи числа - это разрядность и она совпадает с длиной числа. В техническом плане длина числа интерпретируется как длина разрядной сетки. Если алфавит имеет p различных значений, то разряд a_i в числе рассматривается как p -ичная цифра, которой может быть присвоено каждое из p значений.

Каждой цифре a_i данного числа A однозначно соответствует ее количественный (числовой) эквивалент - $K(a_i)$. При любой конечной разрядной сетке количественный эквивалент числа A будет принимать в зависимости от количественных отдельных разрядов значения от $K(A)_{min}$ до $K(A)_{max}$.

Диапазон представления (D) чисел в данной системе счисления - это интервал числовой оси, заключенный между максимальными и минимальными числами, представленными заданной разрядностью (длиной

разрядной сетки):

$$D=K(A)_{(p) \max} - K(A)_{(p) \min}.$$

Существует бесчисленное множество способов записи чисел цифровыми знаками. Однако, любая система счисления, предназначенная для практического использования, должна обеспечивать:

- 1) возможность представления любого числа в заданном диапазоне чисел;
- 2) однозначность представления;
- 3) краткость и простоту записи чисел;
- 4) легкость овладения системой, а также простоту и удобство оперировать ею.

В зависимости от целей применения используют различные системы счисления: 2-ю, 10-ю, 8-ю, 16-ю, римскую, а для исчисления времени - система счисления времени и т.д.

В зависимости от способа записи чисел и способа вычисления их количественного эквивалента системы счисления можно классифицировать следующим образом (рис. 1)

В основном системы счисления строятся по следующему принципу:

$$A_{(p)} = a_n p_n + a_{n-1} p_{n-1} + \dots + a_1 p_1,$$

где $A_{(p)}$ - запись числа в системе с базисом p_i ;

a_i - база или последовательность цифр системы счисления с p_i -чным алфавитом

p_i - базис системы счисления (совокупность весов отдельных разрядов системы счисления). Базис десятичной системы счисления $10^0, 10^1, 10^2, 10^3, \dots, 10^n$.

База системы счисления может быть положительной (0,1,2...9), но

может быть и смешанной (1, $\bar{1}$).

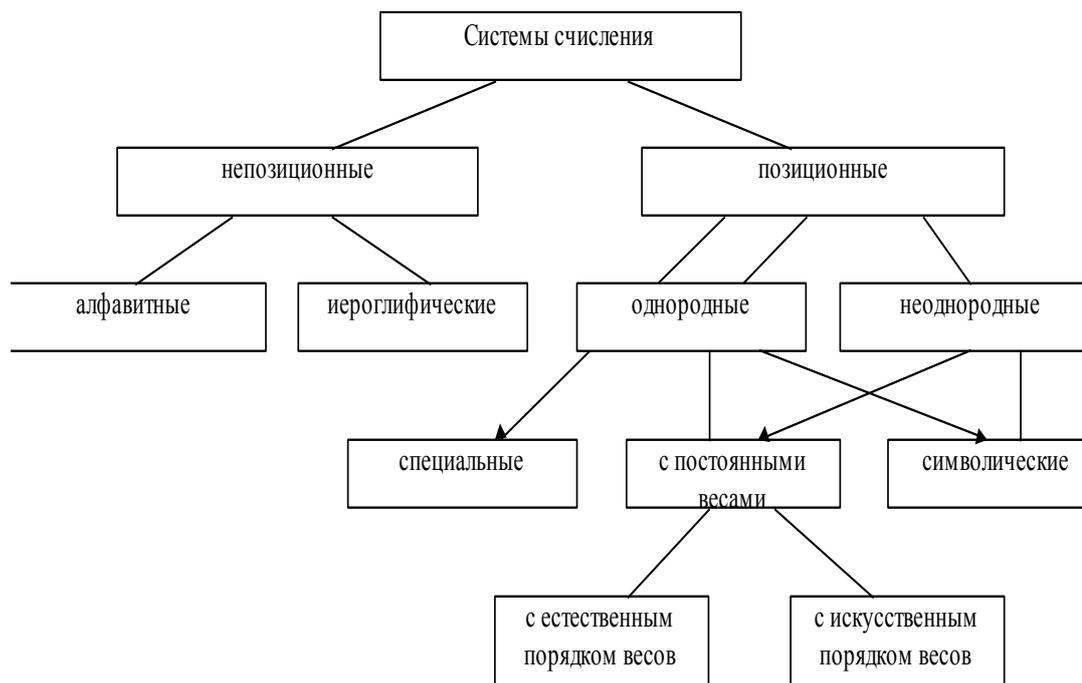


Рисунок. 1- Классификация систем счисления

Основанием системы счисления называется количество различных символов (цифр), используемых в каждом из разрядов для изображения числа в данной системе счисления.

Вес разряда R_i в любой системе счисления - это отношение $R_i = p^i / p_0$.

1.1 Непозиционные системы счисления

Непозиционные системы счисления - это системы счисления, алфавит которых содержит неограниченное количество символов (цифр), причем количественный эквивалент любой цифры постоянен и зависит только от начертания и не зависит от положения в числе. Такие системы строятся по принципу аддитивности, т.е. количественный эквивалент числа определяется как сумма цифр в числе. Наиболее известными представителями непозиционных систем счисления являются иероглифические и алфавитные, в частности, иероглифическая система - римская система счисления. Запись

чисел в алфавитных системах счисления строится по такому же принципу.

К основным недостаткам непозиционных систем счисления можно отнести:

- 1) отсутствие нуля;
- 2) необходимость содержания бесконечного количества символов;
- 3) сложность арифметических действий.

Основное внимание уделим позиционным системам счисления.

1.2 Позиционные системы счисления

Позиционными называются такие системы счисления, алфавит которых содержит ограниченное количество символов, причем значение каждой цифры определяется не только ее начертанием, но и находится в строгой зависимости от позиции в числе. Основное достоинство позиционных систем счисления - удобство выполнения вычислений.

Позиционные системы счисления разделяются на ряд подклассов.

Неоднородные позиционные системы счисления (со смешанным основанием)

В таких системах счисления в каждом разряде количество допустимых символов может быть различным значения не зависят друг от друга и могут принимать любые значения. Примером неоднородной позиционной системы счисления может служить система счисления времени, для которой P_0 - 1 сек, P_1 - 60 сек, P_2 - 60 мин, P_3 - 24 часа, P_4 - 365 суток.

Однородные позиционные системы счисления.

Это частный случай позиционных систем счисления, в них веса отдельных разрядов представляют собой ряд членов геометрической прогрессии со знаменателем p . Поэтому число в однородных системах может быть представлено в общем случае полиномом вида:

$$A_{(p)} = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + \dots + a_{-k} p^{-k},$$

или

$$A = \sum_{i=-k}^n a_i \cdot p^i$$

Основанием однородной позиционной системы может быть любое целое число, так как в определении позиционных систем счисления не наложено никаких ограничений на величину основания. Поэтому возможно бесчисленное множество позиционных систем счисления.

Обычно число в однородной системе счисления записывается в сокращенном виде:

$$A_{(p)} = a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-k},$$

а название системы счисления определяет ее основание: десятичная, двоичная, восьмеричная, и т.д. Для любой позиционной системы счисления справедливо, что ее основание изображается символами 10 в своей системе.

Кодированные системы счисления

Это такие системы, в которых цифры одной системы счисления кодируются при помощи цифр другой системы. Примером может служить двоично-десятичная система с весами (8-4-2-1) или (8-4-2-1+3).

2. Перевод чисел из одной системы счисления в другую

Существует два основных метода перевода чисел из одной системы счисления в другую: табличный и расчетный [2].

Табличный метод прямого перевода основан на сопоставлении таблиц соответствия чисел различных систем счисления. Этот метод очень громоздок и требует очень большого объема памяти для хранения таблиц, но применим для любых систем счисления.

Расчетный метод перевода применим только для позиционных однородных систем счисления.

2.1 Перевод целых чисел из одной позиционной системы счисления в другую

Пусть задано число A в произвольной позиционной системе счисления с основанием L и его необходимо перевести в новую систему счисления с основанием P , т.е. преобразовать к виду:

$$A_{(p)} = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0, \quad (2.1)$$

где $a_i = 0 \div (p-1)$ - база новой системы счисления.

Это выражение можно записать в виде:

$$A = A_1 p + a_0,$$

где $A_1 = (a_n p^{n-1} + a_{n-1} p^{n-2} + \dots + a_2 p^1 + a_1)$ - целая часть частного,

a_0 - остаток от деления A/p , который является цифрой младшего разряда искомого числа.

При делении числа A_1 на p получим остаток a_1 и т.д. Иными словами, если записать выражение (2.1) по схеме Горнера:

$$A = (\dots((a_n p + a_{n-1})p + a_{n-2})p + \dots + a_1)p + a_0,$$

после чего правую часть последовательно разделить на основание новой системы счисления p , то получим коэффициенты:

$$A = A_1 p + a_0$$

$$A_1 = A_2 p + a_1$$

...

$$A_{n-1} = A_n p + a_{n-1}$$

$$A_0 = 0 \cdot p + a_n$$

При этом деление продолжается до тех пор, пока не окажется, что $A_n < p$, $A_{n+1} = 0$

Правило перевода целых чисел из одной позиционной системы счисления в другую формулируется следующим образом:

Чтобы перевести целое число из одной позиционной системы счисления в другую необходимо исходное число последовательно делить на основание новой системы счисления, записанное в исходной системе счисления, до получения частного, равного нулю. Число в новой системе счисления записывается из остатков от деления, начиная с последнего.

Рассмотрим в качестве примера перевод целого числа 138 в двоичную, восьмеричную, шестнадцатеричную системы счисления.

138, 69, 34, 17, 8, 4, 2, 1, 0- частное

0 1 0 1 0 0 0 1 - остаток

138, 17, 2, 0- частное

2 1 2

138, 8, 0

10 8

$$[138]_{10}=[10001010]_2=[212]_8=[8A]_{16}$$

При переводе из двоичной системы счисления в десятичную исходное число необходимо делить на основание новой системы, т.е. на 1010_2 .

Деление выполнить в двоичной системе трудно, поэтому на практике удобнее пользоваться общей записью числа в виде полинома. При переводе двоичных чисел в десятичную систему счисления обычно подсчитывают сумму степеней основания 2, при которых коэффициенты a_i равны 1. Расчеты при этом ведутся в десятичной системе.

2.2 Перевод правильных дробей

Пусть правильную дробь A , заданную в произвольной позиционной системе счисления с основанием L необходимо перевести в новую систему с основанием P , т.е. преобразовать ее к виду:

$$A = a_{-1}p^{-1} + \dots + a_{-k}p^{-k}, \quad (2.2)$$

если, аналогично переводу целых чисел разделить обе части выражения на p^{-1} , т.е. умножить на p , то получим:

$$Ap = a_{-1} + A_1,$$

где $A_1 = a_{-2}p^{-1} + a_{-3}p^{-2} + \dots + a_{-k}p^{-k+1}$ - дробная часть произведения,
 a_{-1} - целая часть результата.

Полученная при этом цифра целой части результата и будет первой цифрой искомого числа. Умножив теперь дробную часть результата на основание новой системы счисления, получим:

$$A_1p = a_{-2} + A_2,$$

где A_2 - дробная часть произведения,

a_{-2} - следующая цифра искомого числа.

Следовательно, при переводе выражение (2.2) представляется по схеме Горнера:

$$A = p^{-1}(a_{-1} + p^{-1}(a_{-2} + \dots + p^{-1}(a_{-k+1} + p^{-1}a_{-k})\dots)).$$

Для перевода правильной дроби из одной позиционной системы счисления в другую ее надо последовательно умножать на основание новой системы счисления до тех пор, пока в новой дроби не будет нужного количества цифр, которое определяется требуемой точностью представления дроби. Правильная дробь в новой системе счисления записывается из целых частей произведений получающихся при последовательном умножении, причем первая целая часть будет старшей цифрой новой дроби.

Рассмотрим в качестве примера перевод правильной дроби 0,536 в двоичную, восьмеричную, шестнадцатиричную системы счисления

$$[0,536]_{10} = [0,10001001]_2 = [0,422335]_8 = [0,8937]_{16}$$

0,	536
	2
<hr/>	
1,	072
	2
<hr/>	
0,	144
	2
<hr/>	
0,	288
	2
<hr/>	
0,	576
	2
<hr/>	
1,	152
	2
<hr/>	
0,	304
	2
<hr/>	
0,	608

0,	536
	8
<hr/>	
4,	288
	8
<hr/>	
2,	304
	8
<hr/>	
2,	432
	8
<hr/>	
3,	456
	8
<hr/>	
3,	648
	8
<hr/>	
5,	184

0,	536
	16
<hr/>	
8,	576
	16
<hr/>	
9,	216
	16
<hr/>	
3,	456
	16
<hr/>	
7,	296

Перевод дроби в общем случае представляет собой бесконечный процесс. Число цифр в новой системе счисления необходимо определять из условия, что точность представления числа в новой системе должна соответствовать точности в исходной системе.

2.3 Перевод неправильных дробей

При переводе неправильной дроби необходимо отдельно перевести целую и дробную части по вышеизложенным правилам и записать число в новой системе счисления, оставив неизменным положение запятой.

2.4 Перевод чисел из системы счисления в систему с кратным основанием

Если основания систем счисления кратны друг другу, т.е. связаны зависимостью: $l = p^m$, то каждая цифра системы счисления с основанием l может быть представлена m цифрами в системе с основанием p .

Следовательно, для того, чтобы перевести число из исходной системы в новую, основание которой кратно основанию исходной системы, достаточно каждую цифру переводимого числа записать при помощи m цифр в новой системе счисления, если основание исходной системы больше основания новой системы счисления. В противном случае каждые m цифр исходного числа необходимо записать при помощи одной цифры в новой системе счисления, начиная для целых чисел с младшего разряда и для правильных дробей - со старшего.

Пример.

$$[0,536]_{10} = [0,100'010'010]_2 = [0,422]_8; \quad [0,1000'1001'0]_2 = [0,89]_{16}$$

$$[138]_{10} = [10'001'010]_2 = [212]_8; \quad [1000'1010]_2 = [8A]_{16}$$

3. Выбор системы счисления для применения в ЭВМ

Очевидно, что непозиционные системы счисления непригодны для применения в ЭВМ в силу своей громоздкости и трудности выполнения арифметических операций.

Из позиционных наиболее удобны однородные. С точки зрения применения в ЭВМ учитываются следующие факторы.

1. Наличие физических элементов, способных изобразить символы системы.
2. Экономичность системы, т.е. количество элементов необходимое для представления многозначных чисел.
3. Трудоемкость выполнения арифметических операций в ЭВМ.
4. Быстродействие вычислительных систем.
5. Наличие формального математического аппарата для анализа и синтеза вычислительных устройств.
6. Удобство работы человека с машиной.
7. Помехоустойчивость кодирования цифр на носителях информации.

Исторически сложилось так, что для применения в ЭВМ была выбрана двоичная система счисления, которая наиболее полно соответствует этим критериям.

В современных универсальных ЭВМ применяются как двоичная, так и десятичная системы счисления. Причем цифры последней кодируются двоичными символами, т. е. речь идет в действительности не о десятичной, а о двоично-десятичной системе счисления. Каждая из отмеченных систем имеет свои достоинства и недостатки, а также свои области применения.

Достоинствами двоичной системы счисления относительно двоично-десятичной являются:

- 1) экономия порядка 20 % оборудования;
- 2) примерно в 1,5 раза более высокое быстродействие;
- 3) упрощение логического построения и значительная экономия

оборудования в схемах управления и во вспомогательных цепях.

Достоинствами двоично-десятичной системы являются:

- 1) отсутствие необходимости перевода исходных данных и результатов расчетов из одной системы в другую;
- 2) удобство контроля промежуточных результатов путем вывода их на индикацию для визуального наблюдения;
- 3) более широкие возможности для автоматического контроля из-за наличия в двоично-десятичном коде избыточных комбинаций.

Двоичную систему счисления применяют в больших и средних ЭВМ, предназначенных для решения научно-технических задач, для которых характерен большой объем вычислений и сравнительно малый объем исходных данных и результатов вычислений. Ее также целесообразно применять в ЭВМ, предназначенных для управления технологическими процессами.

Двоично-десятичную систему счисления применяют для решения экономических задач, которые характеризуются большим объемом исходных данных, сравнительной простотой и малым объемом выполняемых над ними преобразований и большим количеством результатов вычислений. Эту систему целесообразно также применять в калькуляторах, ЭВМ, предназначенных для инженерных расчетов, а также в персональных ЭВМ.

4. Двоичная система счисления

Под двоичной системой счисления понимается такая система, в которой для изображения чисел используются два символа, а веса разрядов меняются по закону 2^{+k} , где k - произвольное целое число. Классической двоичной системой является система с символами 0, 1. Ее двоичные цифры часто называют битами. В общем виде все двоичные числа представляются в виде:

$$A = \sum a_i 2^i, \text{ (i от -к до n)}$$

Чтобы овладеть любой системой счисления, надо уметь выполнять в ней арифметические операции. Арифметические операции в двоичной системе счисления выполняются так же, как и в десятичной в соответствии с таблицами поразрядных вычислений.

Сложение в двоичной системе счисления производится по правилам сложения полиномов. Поэтому при сложении чисел A и B i -й разряд суммы S_i и перенос Π_i из данного разряда в $(i+1)$ разряд будет определяться в соответствии со следующим выражением:

$$a_i + b_i + \Pi_{i-1} = S_i + \Pi_{i+1}$$

a_i	b_i	Π_{i-1}	S_i	Π_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Таблица умножения двух двоичных чисел полностью определяется двумя правилами:

- умножение любого числа на ноль дает в результате ноль,
- умножение любого числа на 1 оставляет его без изменения, т.е. результат равен исходному числу.

4.1 Навыки обращения с двоичными числами

Хотя все правила выполнения операций в двоичной системе счисления очень просты, но тем не менее при работе с двоичными числами из-за отсутствия навыков возникают разного рода неудобства. Ниже приведены некоторые простые приемы, которые позволяют довольно свободно обращаться с двоичными числами.

Таблица 4.1.

n	0	1	2	3	4	5	6	7	8	9	10	11	12
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096

1. Число $100\dots00 = 2^n$.

n нулей

Необходимо знать наизусть десятичные значения чисел, представленных в таблице 4.1.

2. Число $111\dots11 = 2^n - 1$.

n единиц

3. Необходимо знать наизусть десятичные значения двоичных чисел от 0 до 31 включительно. Эти числа в дальнейшем будут называться “малыми числами”.

4. Двоичное число

$$A = \overbrace{a_{n-k+5} a_{n-k+4} a_{n-k+3} a_{n-k+2} a_{n-k+1}}^{\text{малое число}} \overbrace{000\dots000}^{k \text{ нулей равно } a2^k}.$$

Пример. $11011000 = 11011 \times 2^3 = 27 \times 8 = 216.$

Двоичное число

$$A = \overbrace{a_{n-k+5} a_{n-k+4} a_{n-k+3} a_{n-k+2} a_{n-k+1}}^{\text{малое число } a} \overbrace{00\dots00 b_5 b_4 b_3 b_2 b_1}^{\text{малое число } b}$$

k разрядов

Пример. $10110000101 = 1011 \times 2^7 + 101 = 11 \times 128 + 5 = 1413.$

5. Если в n- разрядном числе много единиц и мало нулей, то для определения его количественного эквивалента можно из n разрядного числа, записанного одними единицами, вычесть малое число, в котором разряды со значением 1 соответствуют разрядам исходного числа с нулевым значением и наоборот.

Пример. 11111101001 соответствует

$$11111111111 = 2^{11} - 1$$

$$10110 = 22$$

$$11111101001$$

т.е. $11111101001 = 2048 - 1 - 10110 = 2047 - 22 = 2025.$

6. Чтение двоичных дробей

$$A = \overbrace{0,000\dots001}^{n-1 \text{ нулей}} = 2^{-n}$$

n-1 нулей

$$\text{Дробь } A = \overbrace{0,111\dots111}^{k \text{ единиц}} = 1 - 2^{-k}.$$

k единиц

Двоичная дробь читается по тем же правилам, что и десятичная: разряды справа от запятой читаются как целое число, которое является числителем; знаменатель читается как целое число, являющееся 2^k , причем k - номер младшего разряда справа от запятой.

5. Формы представления двоичных чисел в ЭВМ

Машинное представление числа – это представление числа в разрядной сетке ЭВМ.

Машинное изображение числа условно обозначают $[A]$.

При этом $A=[A]k_A$,

где k_A – масштабный коэффициент, величина которого зависит от формы представления числа в ЭВМ.

Под формой представления числа в ЭВМ понимают свод правил, позволяющий установить взаимное соответствие между записью числа и его количественным эквивалентом.

Если произвольное вещественное число $A'=[A]k_A$, то такое число представлено в разрядной сетке машины точно. Если $A' \neq [A]k_A$, то произвольное вещественное число может быть представлено в машине приближенно или вообще не может быть представлено. При приближенном представлении вещественное число A' заменяется некоторым числом $[A]$, принадлежащим множеству машинных чисел. Множеству машинных чисел принадлежат только числа, кратные двум, так как любые два попарно соседних машинных числа отличаются друг от друга на величину 2^{-n} , где n – количество разрядов.

$$A_{\min} < |A| < A_{\max}$$

Если $|A| < A_{\min}$, такое число называют машинным нулем. Числа, большие чем A_{\max} , не могут быть представлены. В этом случае говорят о переполнении разрядной сетки.

Существует три формы представления чисел в ЭВМ: естественная, с фиксированной запятой и нормальная (с плавающей запятой).

Естественной формой записи числа называется запись числа в виде полинома, представленного в сокращенном виде:

$$A = a_n a_{n-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-k}$$

При этом отсчет весов разрядов ведется от запятой. Запятая ставится на строго определенном месте – между целой и дробной частью числа. Поэтому для каждого числа необходимо указать положение его запятой в одном из разрядов кода, т.е. в общем случае место положения запятой должно быть предусмотрено в каждом разряде. Обычно такую форму представления используют в калькуляторах.

Если место запятой в разрядной сетке машины заранее фиксировано, то такое представление называется представлением с фиксированной запятой (точкой).

В большинстве ЭВМ с фиксированной запятой числа, с которыми оперирует машина, меньше единицы и представлены в виде правильных дробей, т.е. запятую фиксируют перед старшим разрядом числа, причем числа, больше единицы, приводятся к такому виду при помощи масштабного коэффициента K_A . Представление чисел в виде правильных дробей обусловлено необходимостью уменьшить возможность переполнения разрядной сетки машины, т. е. уменьшить опасность потери значащих цифр старших разрядов при выполнении арифметических операций.

Результат умножения никогда не выходит за пределы разрядной сетки, если запятая расположена перед старшим разрядом. Но в этом случае результаты сложения и деления могут выйти за пределы разрядной сетки (при операции сложения — не более чем на один разряд).

Можно было бы оперировать только малыми числами, так как вероятность переполнения при их сложении мала. Однако это приводит к снижению точности представления чисел и точности вычислений. Поэтому всегда стремятся использовать числа, величины которых близки к

максимальному значению. Однако при этом на них накладываются следующие ограничения: 1) абсолютная величина суммы двух чисел должна быть меньше единицы; 2) делитель по абсолютной величине должен быть больше делимого.

В ячейке машины с фиксированной перед старшим разрядом запятой число записывается в разрядную сетку в виде значащей части дроби со своим знаком, т. е. для записи n -значной дроби разрядная сетка должна содержать $n + 1$ разряд.

Разрядная сетка или формат числа в двоичной системе счисления имеет вид:



Здесь n разрядов используют для изображения цифровой части числа и 1 – для знака.

Величины чисел, представляемых в машинах с фиксированной перед старшим разрядом запятой, лежат в пределах:

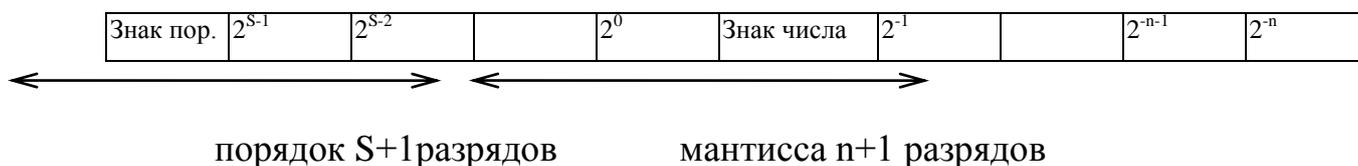
$$2^{-n} \leq |A| \leq 1 - 2^{-n}$$

В этом случае: $|A|_{\min} = 0, \dots, 01 = 2^{-n}$, а $|A|_{\max} = 0,1 \dots 1 = 1 - 2^{-n}$. (Запятая разделяет целую и дробную части).

Начиная с вычислительных машин 2-го поколения, форматы чисел в ЭВМ представляются кратными байту, т. е. $n=8$, или 16, 32.

Во всех рассмотренных форматах могут изображаются числа, которые по своей абсолютной величине меньше 1, что упрощает конструкцию, уменьшает объем оборудования. Недостатком такого представления чисел

Формат числа, представленного в форме с **плавающей запятой**, имеет вид:



В разрядной сетке предусмотрено наличие разряда для фиксации знака мантиссы, который соответствует знаку числа.

Представление числа с плавающей запятой можно проиллюстрировать на следующем примере:

$$987.54 = 10^3 * 0.98754,$$

$$987.54 = 10^4 * 0.098754,$$

$$987.54 = 10^5 * 0.0098754.$$

В целях однозначного представления любого числа введено понятие “нормализованное число”. Нормализованным считается число **A**, мантисса которого удовлетворяет неравенству:

$$2^{-1} \leq |a| \leq 1 - 2^{-n}$$

Другими словами, нормализованным считается то число, у которого старший разряд равен 1.

Диапазон представления порядка числа лежит в пределах:

$$2^S - 1 \geq m \geq -(2^S - 1).$$

Отсюда следует, что диапазон представления чисел для $p = 2$:

минимальное число:

$$|A_{\min}| = 2^{-(2^S - 1)} \cdot 2^{-1} = 2^{-2^S}$$

и максимальное:

$$|A_{\max}| = 2^{(2^s-1)} \cdot (1-2^{-n}) = 2^{2^s-1}$$

Очевидно, что диапазон представления чисел в машинах с плавающей запятой значительно больше, чем в машинах с фиксированной запятой:

$$D = \frac{|A_{\max}|}{|A_{\min}|} \approx \frac{2^{2^s-1}}{2^{-2^s}} = 2^{2^{s+1}-1}$$

Сопоставляя между собой две основные формы представления чисел в ЭВМ, можно прийти к следующим выводам.

Диапазон представления чисел в машинах с фиксированной запятой значительно меньше, чем в машинах с плавающей запятой, а точность зависит от величины исходных чисел. Программирование для машин с фиксированной запятой значительно сложнее, т.к. приходится вводить масштабные коэффициенты, чтобы избежать переполнения масштабной сетки при выполнении арифметических операций.

Однако машины с плавающей запятой конструктивно более сложны, так как необходимо вводить дополнительное оборудование для выполнения операций над порядками чисел, а также предусмотреть операцию нормализации и выравнивания порядков чисел. Время выполнения операций над числами в машине с плавающей запятой больше, чем в машине с фиксированной запятой, что обусловлено необходимостью работы с порядками.

Как и при фиксированной запятой, здесь возможно переполнение разрядной сетки, которое выражается в том, что результат какой-либо операции имеет порядок больше допустимого. Это приводит к аварийной ситуации. При выполнении операций возможно получение чисел, имеющих порядок меньше допустимого и нормализованную мантиссу. Эти числа рассматриваются как машинные нули, так же как и числа, имеющие нулевую мантиссу и допустимый порядок.

Иногда нормальную форму представления чисел называют

полулогарифмической, так как порядок числа p выражен в логарифмической форме.

6. Точность представления чисел в ЭВМ

При решении различных задач требуется различная точность получаемых результатов. Так, при решении инженерных задач достаточно точность до 3—4 десятичных знаков (10—13 двоичных), при решении научных задач — 5—6 десятичных или 16—20 двоичных знаков и при решении особо точных задач — до 50 двоичных разрядов.

При ограниченной длине машинных слов множество чисел, которые можно представить в машине, является конечным. Поэтому представление чисел в ЭВМ, как правило, влечет за собой появление погрешностей, величина которых зависит как от формы представления чисел, так и от длины разрядной сетки.

Точность представления числа характеризуется абсолютной и относительной погрешностями.

Абсолютная погрешность — это разность между истинным значением величины A и ее значением, полученным из машинного изображения $[A]$, т. е.

$$\Delta[A] = A - [A]$$

Усредненная абсолютная погрешность представления чисел в машинах с фиксированной запятой определяется как среднее арифметическое между минимально представимым числом и его минимальной потерей, т. е.

$$\Delta = \frac{A_{\min} + 0}{2} = \frac{2^{-n} + 0}{2} = 2^{-(n+1)}$$

В машинах с фиксированной запятой абсолютная погрешность постоянна и равна половине младшего разряда.

Относительная погрешность представления определяется как

отношение усредненной абсолютной погрешности к самому числу:

$$\varepsilon = \left| \frac{\Delta}{A} \right|.$$

Так как само число с фиксированной запятой меняется в пределах

$$A_{\min} = 2^{-n} \leq |A| \leq 1 - 2^{-n} = A_{\max},$$

то и относительная погрешность является величиной переменной, меняющейся соответственно в пределах

$$\varepsilon_{\min} \leq \varepsilon \leq \varepsilon_{\max}$$

Для машин с фиксированной запятой она определяется следующим образом:

$$\varepsilon_{\min} = \left| \frac{\Delta}{A_{\max}} \right| = \frac{2^{-(n+1)}}{1 - 2^{-n}} \approx 2^{-(n+1)};$$
$$\varepsilon_{\max} = \left| \frac{\Delta}{A_{\min}} \right| = \frac{2^{-(n+1)}}{2^{-n}} \approx 2^{-1}.$$

Таким образом, относительная погрешность для машин с фиксированной запятой зависит от величины числа и колеблется в пределах от $2^{-(n+1)}$ для больших чисел, до 2^{-1} для малых чисел. В машинах с плавающей запятой абсолютная погрешность представления числа определяется следующим образом:

$$\Delta = \Delta_a \cdot 2^m$$

где Δa - погрешность представления мантииссы, которая определяется так же, как абсолютная погрешность представления чисел в машине с фиксированной запятой, т. е. $\Delta a = 2^{-(n+1)}$; m - порядок числа, который изменяется в пределах

$$-(2^S - 1) \leq m \leq +(2^S - 1).$$

Следовательно, в отличие от машин с фиксированной запятой, в машинах с плавающей запятой абсолютная погрешность представления чисел зависит от порядка числа: минимальная при наибольшем отрицательном m и максимальная при наибольшем положительном определяются следующим образом:

$$\Delta_{\min} = 2^{-(n+1)} \cdot 2^{-(2^S - 1)} = 2^{-(n+2^S)};$$

$$\Delta_{\max} = 2^{-(n+1)} \cdot 2^{2^S - 1} = 2^{2^{bl} - n - 2}.$$

Относительная погрешность представления чисел в машинах с плавающей запятой определяется по общему правилу:

$$\varepsilon = \left| \frac{\Delta}{A} \right| = \frac{\Delta a \cdot 2^m}{a \cdot 2^m} = \frac{\Delta a}{a},$$

т. е. ε не зависит от порядка числа и изменяется в пределах

$$\varepsilon_{\min} = \frac{\Delta a}{a_{\max}} = \frac{2^{-(n+1)}}{1 - 2^{-n}} = 2^{-(n+1)};$$

$$\varepsilon_{\max} = \frac{\Delta a}{a_{\min}} = \frac{2^{-(n+1)}}{2^{-1}} = 2^{-n}.$$

Следовательно, в машинах с плавающей запятой, в отличие от машины с фиксированной запятой, относительная погрешность изображения чисел во всем диапазоне представления практически постоянна и для чисел с нормализованной мантиссой зависит от количества разрядов мантиссы: чем их больше, тем меньше погрешность представления.

В некоторых вычислительных средствах информационной единицей являются не отдельные числа, а их блоки или массивы, т. е. последовательности, состоящие из сотен и тысяч чисел. В этих случаях нередко применяется промежуточная форма представления чисел в ЭВМ, так называемое представление с поблочной плавающей запятой, при котором всему массиву чисел присваивается общий порядок и массив считается нормализованным, если хотя бы одно его слово является нормализованным. Естественно, что относительная погрешность представления отдельных элементов массива будет при этом различной. Как и в случае представления с фиксированной запятой, максимальный по абсолютной величине элемент будет представлен с минимальной, в то время как минимальный по абсолютной величине элемент массива — с максимальной относительной погрешностью. Однако это не имеет существенного значения, так как основную информационную нагрузку в этих случаях несут максимальные элементы массивов. Вместе с тем благодаря представлению чисел с поблочной плавающей запятой удается при приемлемой точности вычислений значительно сократить объем оборудования, а главное - время выполнения операции, так как действия над порядками в этом случае выполняются только один раз за время обработки всего массива чисел.

Из этого следует, что нельзя отдать предпочтение какой-либо одной форме представления чисел. Обычно в ЭВМ общего назначения применяют нормальную форму. Этим обеспечивается большой диапазон представления чисел, высокая точность вычислений, простота программирования. Усложнение аппаратуры этих ЭВМ имеет второстепенное значение.

В специализированных ЭВМ чаще применяют фиксированную

или поблочно плавающую запятую, если информация обрабатывается отдельными массивами, так как эти формы обеспечивают простоту конструкции ЭВМ. Диапазон изменения величин известен заранее, масштабные коэффициенты подбираются один раз, требуемая точность вычислений также известна заранее и определяет длину разрядной сетки.

В современных ЭВМ используются обе формы представления чисел. При этом в большинстве случаев формат чисел с фиксированной запятой служит для представления целых двоичных и десятичных чисел и выполнения операций над ними, что, например, необходимо для операций над кодами адресов (операции индексной арифметики).

Вывод

В процессе написания реферата мы ознакомились с:

- с понятием системы счисления;
- классификацией систем счисления;
- переводом чисел из одной системы счисления в другую;
- выбором системы счисления для применения в ЭВМ;
- двоичной системой счисления;
- формами представления двоичных чисел в ЭВМ;
- точностью представления чисел в ЭВМ и др.

Литература

1. Самофалов К.Г., Романкевич А.М., и др. Прикладная теория цифровых автоматов. - Киев. "Вища школа" 1987.
2. Соловьев Г.Н. Арифметические устройства ЭВМ. - М. "Энергия". 1978.
3. Савельев А.Я. Прикладная теория цифровых автоматов - М. "Высшая школа". 1987.
4. Каган Б.М. Электронные вычислительные машины и системы. - М. Энергоатомиздат. 1985.
5. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. - Минск. "Вышэйшая школа". 1980.