Тема 7. Представление информации в ЭВМ.

1. Единицы информации.

Бит - (bit-binary digit - двоичный разряд) наименьшая единица информации - количество её, необходимое для различения двух равновероятных событий.

Пример: бросание монеты, при падении сверху может быть «орел» или «решка». Если загадать что-то одно, то в результате может быть получена информация «ложь» (false) или «истина» (true).

Технически бит - это разряд памяти ЭВМ, где хранится значение 0 или 1.

Байт (byte) - группа из 8-ми бит, обрабатывается как единое целое.

Технически байт - это наименьшая адресуемая единица памяти ЭВМ (ячейка).

Информационно байт - это двоичный код, например, символа (буквы, цифры, знака).

Наглядное представление величины байт: 1 байт = 1 символ

Наиболее популярные соотношения:

- 1 байт = 8 бит
- 1 Кбайт = 1024 байт
- 1 Мбайт = 1024 Кбайт
- 1 Гбайт = 1024 Мбайт
- 1 Тбайт = 1024 Гбайт

Пример записи двоичного кода в байте: 01011001.

Количество таких кодов $2^8=256$.

Наглядное представление сочетаний из 1,2 и 3 бит:

1 бит	2 бит	3 бит
коды: 0 1	коды: 0 0 0 1 1 0 1 1	коды: 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1
количество кодов $2^1=2$	количество кодов $2^2=4$	количество кодов $2^3=8$

Кроме бит и байт в вычислительной технике используется ещё одна специфическая структурная единица информации машинное слово. Это последовательность из двух байтов (16 бит).

2. Системы счисления.

2.1. Позиционные и непозиционные системы счисления.

Системой счисления называется совокупность приемов и правил для записи чисел цифровыми знаками.

Любая предназначенная для практического применения система счисления должна обеспечивать:

- возможность представления любого числа в рассматриваемом диапазоне величин;
- единственность представления (каждой комбинации символов должна соответствовать одна и только одна величина);
- простоту оперирования числами.

Все системы представления чисел делят на *позиционные* и *непозиционные*.

Непозиционная система счисления - система, для которой значение символа не зависит от его положения в числе.

Для их образования используют в основном операции сложения и вычитания. Например, система с одним символом-палочкой встречалась у многих народов. Для изображения какого-то числа в этой системе нужно записать количество палочек, равное данному числу. Эта система неэффективна, так как запись числа получается длинной. Другим примером непозиционной системы счисления является *римская система*, использующая набор следующих символов: I, V, X, L, C, D, M и т. д.

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

В римской системе каждый числовой символ имеет одно и то же значение независимо от его места в записи числа. В этой системе существует отклонение от правила независимости значения цифры от положения в числе. В числах LX и XL символ X принимает два различных значения: +10 – в первом случае и –10 – во втором случае. Примеры: V, XI, XIV, CXXVII.

Позиционная система счисления – система, в которой значение символа определяется его положением в числе: один и тот же знак принимает различное значение.

Например, в десятичном числе 222 первая цифра справа означает две единицы, соседняя с ней – два десятка, а левая – две сотни.

Запись чисел осуществляется справа налево. Крайняя справа цифра означает *единицы*, та же цифра, смещенная на одну позицию влево, означает уже *десятки*, еще левее - *сотни* и т.д.:

$$6421 = \underbrace{6*10^3}_{\text{тысячи}} + \underbrace{4*10^2}_{\text{сотни}} + \underbrace{2*10^1}_{\text{десятки}} + \underbrace{1*10^0}_{\text{единицы}}$$

Количество различных цифр, используемых для представления чисел, называют *основанием системы счисления*.

Для позиционной системы счисления справедливо равенство

$$A_{(q)} = a_n q^n + a_{n-1} q^{n-1} + \ldots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \ldots + a_{-m} q^{-m}, \qquad (1)$$

где **A(q)** – произвольное число, записанное в системе счисления с основанием **q**; **a_i** - коэффициенты ряда (цифры системы счисления); **n**, **m** – количество целых и дробных разрядов.

На практике используют сокращенную запись чисел:

$$A_{(q)} = a_n a_{n-1} \ldots a_1 a_0 a_{-1} \ldots a_{-m}. \qquad (2)$$

Например:
а) в двоичной системе (q=2)
$11010.101_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$;
б) в троичной системе (q=3)
$22120.212_3 = 2 \cdot 3^4 + 2 \cdot 3^3 + 1 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0 + 2 \cdot 3^{-1} + 1 \cdot 3^{-2} + 2 \cdot 3^{-3}$;
в) в шестнадцатиричной системе (q=16)
$A3F.1CD_{16} = A \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0 + 1 \cdot 16^{-1} + C \cdot 16^{-2} + D \cdot 16^{-3}$.

В вычислительной технике используются только позиционные системы счисления:

- двоичная,
- восьмеричная,
- шестнадцатеричная,
- двоично-десятичная.

2.2. Метод перевода чисел.

Числа в разных системах счисления можно представить следующим образом:

$$A_{(q_1)} = a_n q_1^n + a_{n-1} q_1^{n-1} + \ldots + a_1 q_1^1 + a_0 q_1^0 + a_{-1} q_1^{-1} + \ldots + a_{-m} q_1^{-m} =$$

$$= b_{k-1} q_2^{k-1} + \ldots + b_1 q_2^1 + b_0 q_2^0 + b_{-1} q_2^{-1} + \ldots + b_{-s} q_2^{-s} = A_{(q_2)}$$

где $A_{(q)} = a_n q^n + a_{n-1} q^{n-1} + \ldots + a_1 q^1 + a_0 q^0 + a_{-1} q^{-1} + \ldots + a_{-m} q^{-m}$

Значит, в общем виде задачу перевода числа из системы счисления с основанием **q_1** в систему счисления с основанием **q_2** можно представить как задачу определения коэффициентов **b_j** нового ряда, изображающего число в системе с основанием **q_2**. В такой постановке задачу перевода можно решить подбором коэффициентов **b_j**.

2.3. Двоичная система счисления.

Любая информация в современных ЭВМ представляется последовательностью 0 и 1 (бит). Это обусловлено тем, что большинство элементов, из которых состоит ЭВМ, по своей физической природе могут находиться лишь в одном из двух устойчивых состояний «включено» и «выключено». Такие элементы принято называть двухпозиционными. С помощью двухпозиционных элементов легко изображаются разряды двоичного числа. Одно из устойчивых состояний соответствует цифре 0, а другое – 1. В этом отношении двоичная система счисления имеет преимущества перед остальными системами и поэтому оказывается очень удобной для применения в ЭВМ. В двоичной системе легко реализуются арифметические операции, что дает возможность значительно упростить конструкции вычислительных устройств.

В двоичной системе счисления основание системы равно 2 и для представления чисел используются только два символа 0 и 1. Любое число N в двоичной системе представляется в виде суммы степеней основания 2 с соответствующими коэффициентами:

$$N= a_{n-1}\, 2^{n-1} + a_{n-2}\, 2^{n-2} +\ldots+ a_1\, 2^1 + a_0\, 2^0 + = a_{n-1}\, 2^{n-1} +\ldots+ a_{-m}\, 2^{-m} = \sum_{i=-m}^{n-1} a_i 2^i ,$$

где a=0; 1.

Затем с помощью этих коэффициентов число записывается в сокращенной форме.

Например, число 23,625 можно представить в виде суммы:

$23{,}625 = 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1* 2^0 +1*2^{-1} +0*2^{-2} + 1*2^{-3}$

Отсюда может быть получена его запись в двоичной системе счисления:

$23{,}625_{(10)} = 10111{,}101_{(2)}$.

Вариант решения. Смотрим число 23 – какая максимальная степень двойки «умещается» в цифре 23 – 2*2*2*2=16, т.е. 4 степень, остаток = 7 (23-16=7). У 7 максимальная степень двойки = 2, (2*2=4), остаток = 7-4=3 - степень двойки = 1 и остаток = 1

$1 * 2^4$ +(степени = 3 нет, поэтому) + $0*2^3$ (дальше есть степени 2, 1 и 0) $+1*2^2 + 1*2^1 + 1* 2^0$ (дробная часть формируется аналогично, но степени отрицательные) + $1*2^{-1}$ (=0.5) + $0*2^{-2}$ (=0.25 – перебор, значит умножаем на 0) $+1*2^{-3}$ +(1/8 = 0.125 и в сумме с 0.5 дает 0.625)

При разложении числа удобно пользоваться таблицей степеней основания, поэтому этот способ называют табличным.

Таблица степеней

основание / степень	2	8	16
-3	0,125	0,001953	0,000244141
-2	0,25	0,015625	0,00390625
-1	0,5	0,125	0,0625
0	1	1	1
1	2	8	16
2	4	64	256
3	8	512	4096
4	16	4096	65536
5	32	32768	1048576

2.4. Восьмеричная система счисления.

Для ускорения процесса перевода чисел бывает удобнее воспользоваться восьмеричной системой счисления.

Поскольку $8=2^3$, то существует очень простой метод перевода двоичных чисел в восьмеричную систему и наоборот.

Для перевода двоичного числа в восьмеричное надо:

- разбить двоичное число влево и вправо от запятой на группы из 3 цифр (триады);

- И каждой триаде поставить в соответствие его восьмеричный эквивалент.

Восьмеричное число	**Двоичный эквивалент**
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Например:
$N = 1010111011100,10111_2$
Можно записать:
N = (001)(010)(111)(011)(100),(101)(110)
В восьмеричной представлении:
$N = 12734,56_8$

И наоборот, для перехода от восьмеричного к двоичному каждой цифре восьмеричного числа ставят его двоичный эквивалент триаду, затем убирают скобки:

$2543_8 = (010)(101)(100)(011) = 10101100011_2$

2.5. Шестнадцатеричная система счисления

В ЭВМ в качестве единицы информации или объема памяти используют не бит, а байт. Один полубайт соответствует одному разряду шестнадцатеричного числа $2^4 = 16$. Поэтому для более компактного отображения двоичного числа удобнее представлять его в шестнадцатеричной системе счисления, в которой используется 16 символов: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Каждому символу шестнадцатеричного числа ставят в соответствие его двоичный эквивалент - *тетраду*. Для перевода можно использовать таблицу:

Десятичное число	**Шестнадцатеричное число**	**Двоичный эквивалент**
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Например:

N = 1(0101)(1101)(1100)(1011)(1000)$_2$
N =15DC,B8$_{16}$.

2.6. Двоично-десятичная система счисления.

Входная информация в ЭВМ обычно представляется в десятичной системе счисления, а затем по специальным программам переводится в двоичную. Для того, чтобы обрабатывать десятичные числа в машине, их необходимо представить в форме, удобной для машины. С этой целью производится кодирование каждой десятичной цифры с помощью двоичных элементов. Двоично-десятичное представление является наиболее простым представлением, где каждая десятичная цифра представляется своим четырехразрядным двоичным эквивалентом – «тетрадой».

Например:
$237{,}82_{(10)} = (0010)(0011)(0111),(1000)(0010)_{(2\text{-}10)}$

3. Перевод чисел из одной системы счисления в другую.

3.1. Способы перевода.

Переход от двоичного числа к десятичному такой:

Число представляется в форме суммы степеней 2 с соответствующими коэффициентами, которая и вычисляется.

Пример:
$10111_2 = 1*2^4 +0*2^3 +1*2^2 +1*2^1 +1*2^0 =23_{10}$

Перевод из десятичной в двоичную может осуществляться различными способами:

- табличный (используется таблица степеней основания);
- универсальный.

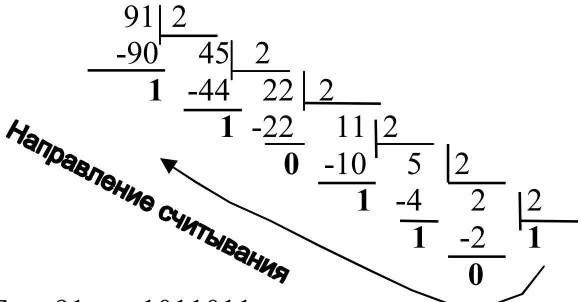
Перевод целых чисел и дробных производится по-разному.

3.2. Перевод целых чисел.

Алгоритм перевода целого числа состоит в делении исходного числа на основании новой системы счисления. Остаток представляет младший разряд числа. Полученное частное вновь делится на основание системы счисления. Остаток дает более старший разряд числа. И так до тех пор, пока не получится частное, меньшее основания новой системы счисления. Следует заметить, что все операции производятся в старой системе счисления.

Пусть, например, необходимо перевести число 9 в двоичную систему счисления.

Последовательно деля его на 2, получаем:



Т.е., $91_{10} = 1011011_2$.

Перевод в восьмеричную систему счисления может быть произведен следующим образом:

91 | 8
-88 | 11 | 8
3 | -8 | 1
3

$91_{10} = 133_8$.

Аналогичен перевод в шестнадцатеричную систему:

91 | 16
-80 | 5
11 = B

$91_{10} = 5B_{16}$.

Пример:

29 : 2 = 14 - остаток 1
14 : 2 = 7 - остаток 0
7 : 2 = 3 - остаток 1
3 : 2 = 1 - остаток 1
1 : 2 = 0 - остаток 1

Двоичное число: 1 1 1 0 1

3.3. Перевод дробных чисел.

Для того, чтобы перевести дробное число из одной системы счисления в другую, его необходимо последовательно умножать на основание новой системы счисления дробные части получаемых произведений. В новой системе дробь записывается в виде последовательности целых частей получаемых произведений.

Пример: $0{,}397_{10} = 0{,}011001_2$.

Направление считывания

0,	397 x 2
0	794 x 2
1	588 x 2
1	176 x 2
0	352 x 2
0	704 x 2
1	408

Операцию умножения можно опускать при записи процесса перевода. Например, $0{,}23_{10} = 0{,}165605_8$

0,	23
1	84
6	72
5	76
6	04
0	64
5	12

3.4. Перевод смешаннных чисел.

Для того, чтобы перевести смешанное число из одной системы счисления в другую, его необходимо:

1. перевести целую часть по правилам перевода целых чисел;
2. перевести дробную часть по правилам перевода дробных чисел.

4. Представление двоичных чисел со знаком.

В двоичной системе есть три способа представления чисел со знаком:

- представление абсолютной величины и знака отдельно (прямой код);
- представление отрицательных чисел в дополнительном коде;
- представление отрицательных чисел в обратном коде.

4.1. Прямой код.

Условно принято обозначать «+» как 0, «-» как 1. Знак располагается в позиции самого старшего разряда. Например:

+10 в прямом коде - это $0|1010_{пк}$;

-10 в прямом коде - это $1|1010_{пк}$.

Этот код достаточно часто используется в машинах.

4.2. Дополнительный код.

Чтобы представить отрицательное число в дополнительном коде, необходимо поменять нули на единицы, а единицы на нули и добавить единицу к самому младшему разряду. Например:

Число -10 записать в дополнительном коде.

+10 в прямом коде это $0|1010_{пк}$

Меняем нули на единицы и единицы на нули:

$0|1010_{пк}$

| | | | |

1|0101

Дополнительный код получается следующим образом:

1|0101

\+ 1

$1|0110_{дк} = -10$

4.3. Обратный код.

Чтобы представить отрицательное число в обратном коде, надо заменить все 1 на 0,а все 0 на 1 и поместить 1 в знаковый разряд.

Пусть то же число -10.

+10 в прямом коде это $0|1010_{пк}$

Меняем нули на единицы и единицы на нули:

$0|1010_{пк}$

| | | | |

1|0101

откуда получаем обратный код: -10

1|0101 $_{ок}$.

5. Представление чисел в ЭВМ.

В вычислительных машинах применяются две формы представления двоичных чисел:

- естественная форма или форма с фиксированной запятой (точкой);
- нормальная форма или форма с плавающей запятой (точкой).

С фиксированной запятой все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Например. В десятичной системе счисления имеются 5 разрядов в целой части числа (до запятой) и 5 разрядов в дробной части числа (после запятой); числа, записанные в такую разрядную сетку, имеют вид:

+00721,35500; +00000,00328; -10301,20260.

Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому не всегда приемлема при вычислениях. В современных ЭВМ естественная форма представления используется как вспомогательная и только для целых чисел.

Для хранения целых неотрицательных чисел отводится одна ячейка памяти (8 бит). Например А2 = 11110000 будет хранится в ячейке памяти следующим образом:

			1	0	0	0	0

Максимальное число – это когда во всех ячейках единицы. Для N- разрядного числа это $2^N - 1$. В восьми битах может храниться число от 0 до 255.

Для хранения целых со знаком отводится две ячейки памяти (16 бит), причем старший (левый) отводится под знак (положительное – 0, отрицательное – 1).

Максимальное положительное (с учетов выделения одного разряда под знак) в N-разрядном представлении равно 2^{N}-1 – 1.

Для представления отрицательных чисел используется дополнительный код, который позволяет заменить арифметическую операцию вычитания сложением, что упрощает работу процессора и увеличивает его быстродействие. Дополнительный код отрицательного числа А, хранящегося в N ячейках, равен $2^N - |A|$. В компьютерной N-разрядной арифметике $2^N = 0$, т.к. действительно запись такого числа состоит из одной единицы и остальные нули, а в N-разрядную ячейку может уместиться только N младших разрядов, т.е. N нулей.

Недостатком представления чисел в формате с фиксированной запятой является небольшой диапазон представления величин, недостаточной для решения математических, физических, экономических и др. задач, где используются как очень малые, так и очень большие числа.

Например, структурно запись числа $-193_{(10)} = -11000001_{(2)}$ в разрядной сетке ПК выглядит следующим образом:

Число с фиксированной запятой формата слово со знаком:

	Знак числ	Абсолютная величина числа														
N	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Число	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

С плавающей запятой каждое число изображается в виде двух групп цифр. Первая группа цифр называется мантиссой, вторая - порядком, причем абсолютная величина мантиссы должна быть меньше 1, а порядок - целым числом. Формат базируется на экспоненциальной форме записи, в которой может быть представлено любое число:

$A = m * q^n$,

где **m** - мантисса числа, **q** - основание системы счисления, **n** - порядок числа.

Для единообразия представления чисел с плавающей запятой используется <u>*нормализованная форма*</u>, при которой мантисса меньше 1, например 555,55 в нормализованном виде = $0{,}55555 * 10^3$ . .

Число в этом формате занимает в памяти машины 4 (обычная точность) или 8 байтов (двойная точность).

Знак порядка

Первый байт 2-й байт 3-й байт ! 4-й байт

<table>
<tr><td>0</td><td>1</td><td>…</td><td>7</td><td>8</td><td>…</td><td>15</td><td>16</td><td>…</td><td>23</td><td>24</td><td>25</td><td>…</td><td>31</td></tr>
<tr><td colspan="10">Мантисса</td><td colspan="4">Порядок</td></tr>
</table>

Знак числа обычно кодируется двоичной цифрой, при этом код 0 означает знак "+", код 1 - знак "-".

Нормальная форма представления имеет огромный диапазон отображения чисел и является основной в современных ЭВМ.

Например, структурно запись числа $-193_{(10)} = -11000001_{(2)}$ в разрядной сетке ПК выглядит следующим образом:

Число с плавающей запятой формата двойное слово:

<table>
<tr><td></td><td>Знак числ</td><td colspan="7">Порядок</td><td colspan="13">Мантисса</td></tr>
<tr><td>N</td><td>31</td><td>3</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>...</td><td>1</td><td>0</td></tr>
<tr><td>Число</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td></td><td>0</td><td>0</td></tr>
</table>