СИСТЕМЫ СЧИСЛЕНИЯ

СИСТЕМЫ СЧИСЛЕНИЯ

Система счисления (СС)- способ представления (записи) чисел с помощью некоторых символов (*цифр*)

ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 1

Единичная система – количество предметов изображалось нанесением черточек (засечек) на твердую поверхность (10-11 тыс. лет до н.э.)

Алфавитные системы (славянская, греческая, финикийская)– числа от 1 до 9 и целые количества десятков, сотен и т.д. обозначались символами

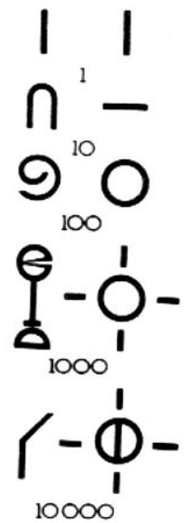
ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 2

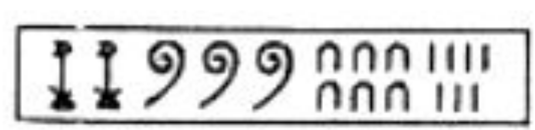
«Особые числа»:

- 1 и 2 – первые числительные, остальные числа получались путем их повторения;
- 3,4 – числительные, связанные с окружающим миром и религией (3 царства, 4 стороны света);
- 5,10,20 – удобство использования для счета (число пальцев);
- 7 – число, связанное с небом (созвездие Б.Медведицы, лунная неделя, планеты-боги, радуга);
- 12 – «дюжина», для счета по пальцам (12 суставов) – время, знаки зодиака;
- 13 – «чертова дюжина», лишнее число
- 60 – время (минуты, секунды), углы
- 100 – «тьма»

ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 3

Первые цифры и запись чисел

Египетская -  - Критская

Пример: 2367 = 

ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 4

Римская система – значение числа равно:

1) сумме значений идущих подряд нескольких одинаковых «цифр» (группы первого вида);

2) разности значений двух «цифр», если слева от большей «цифры» стоит меньшая. В этом случае от значения большей «цифры» отнимается значение меньшей «цифры». Вместе они образуют группу второго вида. Причем левая «цифра» может быть меньше правой максимум на один порядок: так перед L(50) и C(100) из «младших» может стоять только X(10), перед D(500) и M(1000) -только C(100), перед V(5) — только 1(1);

3) сумме значений групп и «цифр», не вошедших в группы первого или второго вида.

Примеры: CDXLIV = (D - C) + (L - X) + (V - I) = 400 + 40 + 4

MCMLXXIV = M + (M - C) + L + (X + X) + (V - I) = 1000 + 900 + 50 + 20 + 4

ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 5

Мультипликативные системы – для записи одинакового числа единиц, десятков, сотен или тысяч применяются одни и те же символы, но после каждого символа пишется название сооветствующего разряда (Вавилон, племена Майя, Индия)

Пример: 323 схематично будет выглядеть так:

3Y 2X 3

(где X – обозначение десятков, Y – обозначение сотен)

ИСТОРИЯ СИСТЕМ СЧИСЛЕНИЯ - 6

Десятичная система – возникла с введением нуля – «О» (от греческого Ouden – «ничто») (Индия, V век н.э.)

ПРИМЕРЫ СИСТЕМ СЧИСЛЕНИЯ

Современная	Египетская (иероглифич.)	Египетская (иератическая)	Вавилонская	Греческая (аттическая)	Греческая (ионическая)	Римская	Древнееврейская	Индейцев майя	Древнекитайская (палочк.)	Древнекит. (иероглифическая)	Индийск. (деванагари)	Арабская (алфавит)	Арабская (современная)	Арабская (гобари)
1	𓏺	\	𒁹	Ι	A	I	א	𝋡	𝍠	一	१	ا	١	1
2	𓏻	\\	𒈫	ΙΙ	B	II	ב	𝋢	𝍡	二	२	ب	٢	2
3	𓏼	\\\	𒐈	ΙΙΙ	Γ	III	ג	𝋣	𝍢	三	३	ج	٣	3
4	𓏽	[illegible]	𒐉	ΙΙΙΙ	Δ	IIII	ד	𝋤	𝍣	四	४	د	٤	4
5	𓏾	[illegible]	𒐊	Γ	E	V	ה	𝋥	𝍤	五	५	ه	٥	5
6	𓏿	[illegible]	𒐋	ΓΙ	F	VI	ו	𝋦	𝍥	六	६	و	٦	6
7	𓐀	[illegible]	𒐌	ΓΙΙ	Z	VII	ז	𝋧	𝍦	七	७	ز	٧	7
8	𓐁	[illegible]	𒐍	ΓΙΙΙ	H	VIII	ח	𝋨	𝍧	八	८	ح	٨	8
9	𓐂	[illegible]	𒐎	ΓΙΙΙΙ	Θ	IX	ט	𝋩	𝍨	九	९	ط	٩	9
10	𓎆	[illegible]	𒌋	Δ	I	X	י	𝋪	𝍩	十	१०	ي	١٠	10
20	𓎇	[illegible]	𒎙	ΔΔ	K	XX	כ	𝋡 𝋠	𝍪	二十	२०	ك	٢٠	20

ПОЗИЦИОННЫЕ И НЕПОЗИЦИОННЫЕ СС

В *непозиционных* системах счисления вес цифры не зависит от позиции, которую она занимает в числе (XXXL)

В *позиционных* системах счисления вес каждой цифры изменяется в зависимости от ее позиции в последовательности цифр, изображающих число (243, 324, 432)

ПОЗИЦИОННЫЕ СС

Количественное значение (*величина*) цифры определяется ее видом и положением в записи числа

Основание системы счисления – количество различных цифр, используемых для представления числа

Основание 10 у привычной десятичной системы счисления (десять пальцев на руках). Алфавит: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.
Основание 60 придумано в Древнем Вавилоне: деление часа на 60 минут, минуты — на 60 секунд, угла — на 360 градусов.
Основание 12 распространили англосаксы: в году 12 месяцев, в сутках два периода по 12 часов, в футе 12 дюймов.

ПРЕДСТАВЛЕНИЕ ЧИСЛА - 1

Позиция цифры в числе называется *разрядом*.

$\mathbf{A}_q = \mathbf{a}_{n-1}\times q^{n-1} + \ldots + \mathbf{a}_1\times q^1 + \mathbf{a}_0\times q^0 + \mathbf{a}_{-1}\times q^{-1} + \ldots + \mathbf{a}_{-m}\times q^{-m}$,
где

q — основание системы счисления (*количество используемых цифр*)

$\mathbf{A}_q$ — число в системе счисления с основанием q

a — цифры многоразрядного числа A_q

n (m) — количество целых (дробных) разрядов числа A_q

Пример:

$$\overset{2}{2}\overset{1}{3}\overset{0}{9},\overset{-1}{4}\overset{-2}{5}_{10} = 2\times10^2 + 3\times10^1 + 9\times10^0 + 4\times10^{-1} + 5\times10^{-2}.$$

$a_2\ a_1\ a_0,\ a_{-1}\ a_{-2}$

ПРЕДСТАВЛЕНИЕ ЧИСЛА - 2

Одно и то же число может быть представлено в различных системах счисления (с разными основаниями)

$$101111_2 \leftarrow [47] \rightarrow 57_8$$
$$1202_3 \leftarrow [47] \rightarrow 47_{10}$$
$$223_4 \leftarrow [47] \rightarrow 2F_{16}$$

[.] – значение числа. Для простоты понимания значение числа всегда указывается в десятичной СС

ПЕРЕВОД ЧИСЛА $N_p \Rightarrow N_q$

Перевод числа из одной СС в другую осуществляется в два этапа:

1) переводится целая часть числа;
2) переводится дробная часть числа.

ПЕРЕВОД ЧИСЛА $N_p \Rightarrow N_q$.

(правило перевода целой части числа)

Для перевода целого числа N_p в число N_q необходимо N_p делить на основание q (по правилам, принятым в $СС_p$) до получения целого остатка, меньшего q. Полученное частное снова необходимо делить на основание q до получения целого остатка, меньшего q и т.д. до тех пор, пока последнее частное не будет меньше q.

Число N_q представится в виде упорядоченной последовательности цифр $СС_q$ (остатков от деления) в порядке, обратном получению, причем старшую цифру числа N_q даст последнее частное

ПЕРЕВОД ЧИСЛА $N_p \Rightarrow N_q$.

(правило перевода дробной части числа)

Перевод правильной дроби N_p в число N_q заключается в последовательном умножении дроби N_p на основание q (по правилам, принятым в $СС_p$), причем перемножению подвергается только дробная часть.

Дробь N_q представится в виде упорядоченной последовательности целых частей произведений в порядке их получения.

В общем случае при переводе может возникать погрешность вследствие конечности разрядной сетки. Если требуемая точность перевода есть q^{-k}, то число указанных произведений должно быть равно k.

ПЕРЕВОД ЧИСЛА $N_p \Rightarrow N_q$.

(упражнения)

$349_{10} \rightarrow ?_4$

$0{,}41_{10} \rightarrow ?_2$

$24{,}18_{10} \rightarrow ?_3$

$534_{10} \rightarrow ?_{16}$

...

$3A_{16} \rightarrow ?_4$

$2{,}7_8 \rightarrow ?_{10}$

$3{,}7_8 \rightarrow ?_2$

...

ПЕРЕВОД ЧИСЛА $N_8 \Rightarrow N_2$, $N_{16} \Rightarrow N_2$

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему: каждую цифру заменить эквивалентной ей двоичной *триадой* (тройкой цифр) или *тетрадой* (четверкой цифр).

Примеры:

$5371_8 = 101\ 011\ 111\ 001_2$;

5 3 7 1

$1A3F_{16} = 1\ 1010\ 0011\ 1111_2$

1 A 3 F

ПЕРЕВОД ЧИСЛА $N_2 \Rightarrow N_8$, $N_2 \Rightarrow N_{16}$

Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на *триады* (для восьмеричной) или *тетрады* (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Примеры:

$1101010000111_2 = 1\ \ 5\ \ \ 2\ \ \ 0\ \ \ 7_8$;

1 101 010 000 111

$110111000001101_2 = \ \ 6\ \ \ \ E\ \ \ \ 0\ \ \ \ D_{16}$

110 1110 0000 1101

ДВОИЧНАЯ АРИФМЕТИКА

Таблица сложения

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = 10

Таблица вычитания

0 – 0 = 0
1 – 0 = 1
1 – 1 = 0
10 – 1 = 1

Таблица умножения

0 × 0 = 0
1 × 0 = 0
1 × 1 = 1

```
  1 1 0 1 1
 1 0 1 1 0 1
 -----------
1 0 0 1 0 0 0
```

```
1 0 0 1 0 0 0
 -----------
 1 0 1 1 0 1
  1 1 0 1 1
```

```
1 1 0 1 0 1 0 0 1 | 1 0 0 0 1
1 0 0 0 1         |----------
  1 0 0 1 1         1 1 0 0 1
  1 0 0 0 1
      1 0 0 0 1
      1 0 0 0 1
      0 0 0 0 0
```

```
          1 1 0 0 1
          1 0 0 0 1
 -------------------
          1 1 0 0 1
        0 0 0 0 0 _
      0 0 0 0 0 _ _
    0 0 0 0 0 _ _ _
  1 1 0 0 1 _ _ _ _
 -------------------
  1 1 0 1 0 1 0 0 1
```

КОДЫ И КОДИРОВАНИЕ ЧИСЕЛ -1

Код – это правило, описывающее отображение одного набора знаков в другой набор знаков.

Кодом также называют *множество образов*, получаемых при этом отображении.

Примеры:

[45] = 101101_2 (двоичное кодирование и двоичный код)

«A», «B», «1», «a», «+»,... = 65, 66, 49, 97, 43,..

(ASCII-код)

КОДЫ И КОДИРОВАНИЕ ЧИСЕЛ -2

Примеры кодов:

-*код Бэкона* (1561-1626) – каждый символ представляется комбинацией из пяти символов «А» и «В»:

a = AAAAA, k = ABAAB, t = BAABA

-*код Грея* – соседние слова отличаются не более, чем в одном разряде:

N	Двоичный код	Код Грея	N	Двоичный код	Код Грея
0	000	000	4	100	110
1	001	001	5	101	111
2	010	011	6	110	101
3	011	010	7	111	100

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ -1

- ЧИСЛА
  - С фиксированной точкой
    - Беззнаковые (*unsigned*)
    - Знаковые (*signed*)
  - С плавающей точкой

В языках программирования высокого уровня (С/С++, Pascal) некоторые типы «по умолчанию» являются знаковыми (например, `int` ≡ `signed int`), а некоторые – беззнаковыми (например, `char` ≡ `unsigned char`).

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ -2

Числа с плавающей точкой:
при выполнении операций местоположение точки в записи числа изменяется таким образом, чтобы слева от точки оставался один разряд, а смещение точки описывалось экспонентой:

$$\begin{array}{r} 4.8 * 10^1 \\ 5.6 * 10^1 \\ \hline 10.4 * 10^1 \end{array} \rightarrow \mathbf{1.04 * 10^2}$$

Формат (IEEE-754)	*S*	*E*	*M*	*Диапазон чисел:*
Single Precision (32)	1 (31)	8 (30 – 23)	23 (22-00)	от ± $\sim 10^{-44.85}$ до $\sim 10^{38.53}$
Double Precision (64)	1 (63)	11 (62 - 52)	52 (51-00)	от ± $\sim 10^{-323.3}$ до $\sim 10^{308.3}$

Достоинства: большой диапазон обрабатываемых значений

Недостатки: сложность в реализации устройства обработки

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ -3

Числа с фиксированной точкой: местоположение точки в записи числа не изменяется.
Точка, разделяющая целую и дробную часть, *теоретически* может располагаться между любыми двумя разрядами.
На *практике* для *целых* ФЗ-чисел считается, что точка расположена слева от самого младшего разряда: 01011001**.**

$$\begin{array}{r} 48. \\ 56. \\ \hline 104. \end{array}$$

Достоинства: простота реализации устройства обработки, высокая точность, интуитивная понятность

Недостатки: малый диапазон возможных значений

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ БЕЗ ЗНАКА

Представление беззнакового (unsigned) числа соответствует его записи в заданной системе счисления (обычно двоичной или шестнадцатеричной)

Машинное представление числа (беззнакового или знакового) выполняется с учетом разрядности *n* машинного слова

Диапазон представления беззнаковых чисел:

$$0 \le x \le 2^n-1$$

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ СО ЗНАКОМ

Знаковые (signed) числа представляются в ЭВМ:

- в прямом коде;
- в обратном коде;
- в дополнительном коде

Для обозначения знака числа в этих кодах выделяется специальный знаковый разряд, в котором записывается «0» для положительного числа и «1» для отрицательного числа.

Знаковый разряд всегда располагается слева от цифровых разрядов.

ПРЯМОЙ КОД

Число представляется в виде его абсолютного значения и кода знака

$$[x]_{\text{пр}} = \begin{cases} |x|, \ \text{если } x \geq 0; \\ 2^{n-1} + |x|, \ \text{если } x \leq 0 \end{cases}$$

Диапазон представления:

$$1-2^{n-1} \leq x \leq 2^{n-1} - 1$$

Представления «0»:

$$[+0]_{\text{пр}} = 0'00...0_2$$

$$[-0]_{\text{пр}} = 1'00...0_2$$

ПРЯМОЙ КОД (пример)

Представить в прямом коде для n=5, n=8

x = [13] x = [-13]

$$/x/ = /13/_{10} = 1101_2 = D_{16}$$

<u>n=5:</u>	$[13] = 0'1101_{пр,2}$	$[-13] = 1'1101_{пр,2}$
<u>n=8:</u>	$[13] = 0'0001101_{пр,2} = 0D_{пр,16}$	$[-13] = 1'0001101_{пр,2} = 8D_{пр,2}$

ОБРАТНЫЙ КОД

Обратный код положительного числа $x \geq 0$ содержит «0» в старшем знаковом разряде и обычное представление x в остальных разрядах.

Если $x \leq 0$, то знаковый разряд содержит «1», а остальные разряды содержат инвертированные значения

$$[x]_{\text{обр}} = \begin{cases} |\,x\,|, & \text{если x} \geq 0; \\ (2^{n-1} - 1) - |\,x\,|, & \text{если x} \leq 0 \end{cases}$$

Диапазон представления:

$$1 - 2^{n-1} \leq x \leq 2^{n-1} - 1$$

Представления «0»:

$$[+0]_{\text{обр}} = 0'00\ldots0_2$$

$$[-0]_{\text{обр}} = 1'11\ldots1_2$$

ОБРАТНЫЙ КОД (пример)

Представить в обратном коде для n=5, n=8

x = [13] x = [-13]

$$/x/ = /13/_{10} = 1101_2 = D_{16}$$

<u>n=5:</u> [13] = 0’1101$_{пр,2}$ = = 0’1101$_{обр,2}$	[-13] = 1’1101$_{пр,2}$ = = 1’0010$_{обр,2}$
<u>n=8:</u> [13] = 0’0001101$_{пр,2}$ = 0D$_{пр,16}$ = = 0’0001101$_{обр,2}$ = 0D$_{обр,16}$	[-13] = 1’0001101$_{пр,2}$ = 8D$_{пр,2}$ = = 1’1110010$_{обр,2}$ = F2$_{обр,16}$

ПРАВИЛО СЛОЖЕНИЯ В ОБРАТНОМ КОДЕ

Коды слагаемых суммируются, включая знаковый разряд, с циклическим (круговым) переносом.

Результат верен, если не произошло переполнение.

Переполнение происходит тогда, когда перенос в знаковый разряд (C_s) не равен переносу из знакового разряда (C_{s+1})

ДОПОЛНИТЕЛЬНЫЙ КОД

Дополнительный код положительного числа $x \geq 0$ содержит «0» в старшем знаковом разряде и обычное представление x в остальных разрядах (совпадает с прямым и обратным).

Если $x \leq 0$, то знаковый разряд содержит «1», а остальные разряды содержат дополнение модуля исходного числа до 2^{n-1}.

$$[x]_{\text{доп}} = \begin{cases} |x|, & \text{если } x \geq 0; \\ 2^{n-1} - |x|, & \text{если } x < 0 \end{cases}$$

Диапазон представления:

$$-2^{n-1} \leq x \leq 2^{n-1} - 1$$

Представление «0»:

$$[0]_{\text{обр}} = 0'00...0_2$$

ДОПОЛНИТЕЛЬНЫЙ КОД (пример)

Представить в дополнительном коде для $n=5$, $n=8$

$x = [13]$ $\qquad$ $x = [-13]$

$/x/ = /13/_{10} = 1101_2 = D_{16}$

<u>$n=5$:</u>	$[13] = 0'1101_{пр,2} =$ $= 0'1101_{обр,2} =$ $= 0'1101_{доп,2}$	$[-13] = 1'1101_{пр,2} =$ $= 1'0010_{обр,2} =$ $= 1'0011_{доп,2}$
<u>$n=8$:</u>	$[13] = 0'0001101_{пр,2} = 0D_{пр,16} =$ $= 0'0001101_{обр,2} = 0D_{обр,16} =$ $= 0'0001101_{доп,2} = 0D_{доп,16}$	$[-13] = 1'0001101_{пр,2} = 8D_{пр,2} =$ $= 1'1110010_{обр,2} = F2_{обр,16} =$ $= 1'1110011_{доп,2} = F3_{доп,16}$

ПРАВИЛО СЛОЖЕНИЯ В ДОПОЛНИТЕЛЬНОМ КОДЕ

Коды слагаемых суммируются, включая знаковый разряд. Перенос (если он есть) отбрасывается.

Результат верен, если не произошло переполнение.

Переполнение происходит тогда, когда перенос в знаковый разряд (C_s) не равен переносу из знакового разряда (C_{s+1})

УВЕЛИЧЕНИЕ РАЗРЯДНОСТИ ЧИСЕЛ ПРИ ПРИСВАИВАНИИ

Для беззнаковых (unsigned) чисел поле расширения в переменной-результате заполняется нулями

Для знаковых (signed) чисел поле расширения в переменной-результате заполняется знаковым битом

В языках высокого уровня способ расширения выбирается и реализуется компилятором по типу данных автоматически

В Ассемблере программист самостоятельно выбирает способ реализации расширения разрядности переменной

УМНОЖЕНИЕ ЦЕЛОГО ЧИСЛА НА КОНСТАНТУ ПОСРЕДСТВОМ СДВИГОВ

Сдвиг *беззнаковых* (*unsigned*) или *знаковых* (*signed*) числа **влево** на *n* двоичных разрядов приводит к его **умножению** на 2^n

Если переменную *V* необходимо умножить на константу *C*, то константа *C* представляется в виде суммы степеней числа 2, а результат умножения записывается как сумма сдвигов числа *V* на показатели степеней.

Пример: *V = V * 25;* $C = 25 = 11001_2 = 16+8+1 = 2^4 + 2^3 + 2^0$

V = *V*(16+8+1) = V*16 + V*8 + V =* ***(V<<4) + (V<<3) + V***

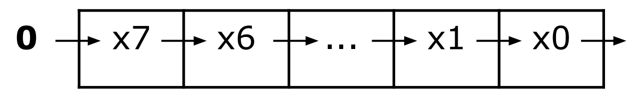
Достоинства: сдвиги и сложения выполняются быстрее, чем умножение

Недостатки: формула зависит от конкретного значения C, т.е. нельзя таким способом перемножить две переменных.

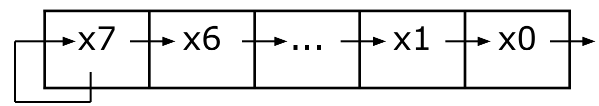
ДЕЛЕНИЕ ЦЕЛОГО ЧИСЛА НА 2^n ПОСРЕДСТВОМ СДВИГОВ

Сдвиг *беззнаковых* (*unsigned*) или *знаковых* (*signed*) числа **вправо** на *n* двоичных разрядов приводит к его **делению** на 2^n

Для сдвига вправо *беззнаковых (unsigned)* чисел используется *логический* сдвиг:



Для сдвига вправо *знаковых (signed)* чисел используется *арифметический* сдвиг:



Пример: *V = V / 16;* C = 16 = $10000_2 = 2^4$ ***V = (V>>4);***

Достоинства: сдвиги выполняются быстрее, чем деление. Компилятор автоматически выбирает правильную команду сдвига (по типу данных)

Недостатки: фактор сдвига зависит от конкретного значения *C*, таким способом нельзя выполнить деление на переменную или число $\neq 2^n$