Представление числовой информации в ПК

Диденко В.В.

Представление числовой информации в ПК

Формат с фиксированной точкой

(только целые)

Формат с плавающей точкой

(вещественные)

Представление чисел в формате с фиксированной запятой

- **Целые числа.** Каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа, запятая находится справа после младшего разряда.

- Для хранения целых неотрицательных чисел отводится одна ячейка памяти (8 бит), 2 ячейки памяти (16 бит).

- В **k-разрядной** ячейке может храниться 2^k различных значений целых чисел. (из 2х - 2^2=4, из 3х - 2^3=8, из 4х – 2^4=16 и т.д.)

  Например, число A_2=11110000 будет храниться в ячейке следующим образом:

1	1	1	1	0	0	0	0

- Максимальное значение достигается, когда во всех ячейках будут 1.

Для **n-разрядного** представления 2^n-1

Представление чисел в формате с фиксированной запятой

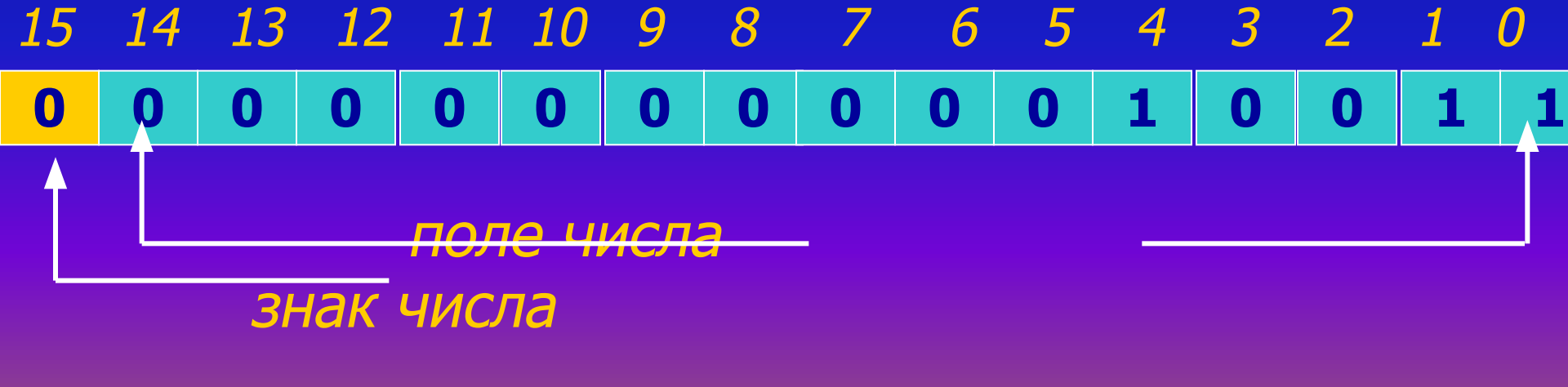
- Целые числа без знака в двухбайтовом формате могут принимать значения

От 0 до 2^{16}-1 (до 65535)

- Целые числа со знаком в двухбайтовом формате могут принимать значения

От -2^{15} до $+2^{15}$-1 (от -32768 до + 32767)

- Например, 19 (10011_2) в 16-разрядном представлении в памяти ПК записывается так:



Алгоритм внутреннего представления целого положительного числа N, хранящегося в k-разрядном машинном слове:

1. Перевести число N в двоичную СС

2. Полученный результат дополнить слева незначащими нулями до k разрядов.

Пример 1. Получить внутреннее представление целого положительного числа 1607 в 2-х байтовой ячейке.

Решение

1. $N = 1607_{10} = 11001000111_2$

2. Внутреннее представление числа:

   0000 0110 0100 0111

3. Шестнадцатеричная форма внутреннего представления числа:

   0647

Представление целого отрицательного числа

Формирование дополнительного кода

1. **Дополнительный код позволяет заменить арифметическую операцию вычитания операцией сложения.**
2. **Дополнительный код отрицательного числа A, хранящегося в n ячейках, равен**

$$2^n - |A|$$

Дополнение модуля отрицательного числа A до 0

Алгоритм внутреннего представления целого отрицательного числа N, хранящегося в k-разрядном машинном слове:

1. **Получить внутреннее представление положительного числа N.**
2. **Получить обратный код этого числа**
   заменой 0 на 1 и 1 на 0.
3. **К полученному числу добавить 1.**

Данная форма представления целого отрицательного числа называется дополнительным кодом.

код отрицательного числа - 2002 для 16-разрядного компьютерного представления.

Прямой код модуля	$\lvert -2002_{10} \rvert$	$0000011\ 111010010_2$
Обратный код	Инвертирование	1111100000101101_2
	Прибавление единицы	1111100000101101_2 + 0000000000000001_2
Дополнительный код		1111100000101110_2

Примеры. **Выполнить арифметическое действие $3000_{10} - 5000_{10}$ в 16-ти разрядном компьютерном представлении.**

1) ***Представим положительное число в прямом, а отрицательное число в дополнительном коде:***

Десятичное число	Прямой код	Обратный код	Дополнительный код
3000	0000101110111000		
-5000	0001001110001000	1110110001110111	1110110001110111 +0000000000000001 1110110001111000

2) Сложим прямой код положительного числа с дополнительным кодом отрицательного числа. Получим результат в дополнительном коде:

3000-5000			1111100000110000

3) Переведем полученный дополнительный код в десятичное число:

1. Инвертируем дополнительный код: 0000011111001111
2. Прибавим к полученному коду 1 и получим модуль отрицательного числа:

```
  0000011111001111
+ 0000000000000001
  0000011111010000
```

3. Переведем в десятичное число и припишем знак отрицательного числа: -2000.

Достоинства представления чисел в формате с фиксированной точкой

- *Простота*
- *Наглядность представления чисел*
- *Простота алгоритмов реализации арифметических операций*

Недостатки представления чисел в формате с фиксированной точкой

- *небольшой диапазон представления величин, недостаточный для решения математических, экономических и др. задач.*

Задания

- **Компьютер работает только с целыми положительными числами. Каков диапазон изменения чисел, если для представления числа в памяти компьютера отводится 1 байт?**

От 0 до 255

- **Каков диапазон изменения целых чисел (положительных и отрицательных), если в памяти ПК для представления целого числа отводится 1 байт?**

От -128 до 128

- **Компьютер работает только с целыми положительными числами. Каков диапазон изменения чисел, если для представления числа в памяти компьютера отводится 4 байт?**

От 0 до $2^{32}-1$

- **Каков диапазон изменения целых чисел (положительных и отрицательных), если в памяти ПК для представления целого числа отводится 4 байта?**

От -2^{31} до $2^{31}-1$

Индивидуальная работа

- **Получить двоичную форму внутреннего представления целого числа в 2-х байтовой ячейке.**
- **Получить шестнадцатеричную форму внутреннего представления целого числа в 2-х байтовой ячейке.**
- **По шестнадцатеричной форме внутреннего представления целого числа в 2-х байтовой ячейке восстановить само число.**

Стр. 137 (И. Семакин, Задачник-практикум 1)

Представление чисел в формате с плавающей запятой

- Базируется на экспоненциальной форме записи числа:

$$A = m * q^n$$

m - мантисса числа
Q – основание СС
N – порядок числа

- Для единообразия представления чисел используется нормализованная форма:
  1/n <= |m| <=1 ***(правильная дробь и после запятой цифра, отличная от нуля.)***

- Пример. Преобразуйте число 555,55, записанное в естественной форме, в экспоненциальную форму с нормализованной мантиссой: $555,55 = 0,55555 * 10^3$

Нормализованная мантисса: 0,55555
Порядок: n = 3

Пример

- Число в форме с плавающей точкой занимает в памяти компьютера 4 или 8 байт.
- При записи числа с плавающей точкой выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.
- Любое вещественное число в современных ПК представляется в экспоненциальной форме с нормализованной мантиссой. При этом мантисса является правильной двоичной дробью, а порядок — целым числом.
- Преобразуйте десятичное число **888,888**, записанное в естественной форме, в экспоненциальную форму с нормализованной мантиссой.

$$888{,}888 = 0{,}888888 \times 103$$

Нормализованная мантисса $m_A = 0{,}888888$, порядок P=3.

Представление чисел в формате с плавающей запятой

- Занимает в памяти ПК **4** *(обычная точность)* или 8 байтов *(двойная точность)*
- Выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.

1-й байт *2-й байт* *3-й байт* *4-й байт*

Представление чисел в формате с плавающей запятой

- в старшем бите 1-го байта хранится знак числа

0 – обозначает плюс, 1 – минус;

- 7 бит содержат **машинный порядок;**
- в следующих трех байтах, хранятся **значащиеся цифры мантиссы** (24 разряда).

В оставшихся семи разрядах 1-го байта

Помещается двоичное число в диапазоне ***от 0000000 до 1111111***, т.е. машинный порядок изменяется в диапазоне от ***0 до 127 (всего 128).***

Порядок м.б. положительным и отрицательным, разумно 128 значений делить поровну: от -64 до +63.

Машинный порядок смещен относительно математического и имеет только положительные значения.

Смещение выбирается так, чтобы минимальному математическому значению соответствовал нуль.

$M_P = P + 64$

Машинный порядок	десятичное число		Математический порядок	Смещение
0000000	0	=	- 64	+64
. . .				
0001010	10	=	- 54	+64
. . .				
1111111	127	=	63	64

Алгоритм записи внутреннего представления вещественного числа

1. Перевести модуль числа в двоичную СС с 24 значащими цифрами.
2. Нормализовать двоичное число.
3. Найти машинный порядок в двоичной СС.
4. Учитывая знак числа, записать его в 4-х байтовом машинном слове.

Пример 1 . Записать внутреннее представление числа 250,1875 в форме с плавающей точкой.

Решение

1. Переведем в двоичную СС:

$$50{,}1875_{10} = 11111010,\ 0011000000000000_2$$

2. Запишем в форме нормализованного двоичного числа: 0, 11111010 0011000000000000 * 10_2^{1000}

(мантисса, основание СС $2_{10}=10_2$ и порядок $8_{10}=1000_2$)

3. Вычислим машинный порядок в двоичной СС:

$$Мр_2 = 1000 + 100\ 0000 = 100\ 1000$$

3. Запишем число в 4-х байтовой ячейке:

Шестнадцатеричная форма 48FA3000

Пример 2 . По шестнадцатеричной форме внутреннего представления числа в форме с плавающей точкой **CC98811000** восстановить число.

Решение

1. Перейдем к двоичному представлению числа:

1100 1001 1000 0001 0001 0000 0000 0000

2. В старшем разряде с номером 31 записана 1, значит получен код отрицательного числа. Получим порядок числа:

$$p=1001001_2 - 1000000_2 - 10001_2 = 9_{10}$$

3. Запишем в виде нормализованного дв. числа с плавающей точкой с учетом знака числа: $-0,10000001000100000000000 \cdot 2^{1001}$
4. В двоичной системе СС число имеет вид: $-100000010,001_2$
5. Переведем число в десятичную СС:

$$-100000010,001_2 = -(1*2^8+1*2^1+1*^2-3) = -258,125_{10}$$

Представление чисел в формате с плавающей запятой

- Занимает в памяти ПК **4** *(обычная точность)* или 8 байтов *(двойная точность)*
- Выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.
- Максимальное значение порядка числа:

$$1111111^2 = 127_{10}$$

- Максимальное значение числа составляет:

$$2^{127} = 1,\ 70141183460469231731687303715884 * 10^{38}$$

- Максимальное значение положительной мантиссы равно:

$$2^{23} - 1 \sim 2^{23} = 2^{(10*2,3)} \sim 1000^{2,3} = 10^{(2,3*3)} \sim 10^7$$

Максимальное значение чисел обычной точности вычислений составляет $1,701411 * 10^{38}$

Задания

1. Заполните таблицу:

Десятичное число	Прямой код	Обратный код	Дополнительный код
-50			
-500			

2. Определите диапазон представления целых чисел со знаком (2 байта памяти) в формате с фиксированной запятой.

3. Определите максимальное число и его точность для формата чисел двойной точности, если для хранения порядка и его знака отводится 11 разрядов, а для хранения мантиссы и ее знака -53 разряда.

ОТВЕТЫ

2. **От - 32 768 до 32 767.**

3. Максимальное значение чисел двойной точности с учетом возможной точности вычислений составит **$8{,}98846567431157 * 10^{307}$**
(количество значащих цифр десятичного числа в данном случае ограничено 15-16 разрядами).

1.

Десятичные числа	Прямой код	Обратный код	Дополнительный код
-50	0000000000110010	1111111111001101	1111111111001110
-500	0000000111110100	1111111000001011	1111111000001100