РОСЖЕЛДОР
Федеральное государственное бюджетное
образовательное учреждение высшего образования
Ростовский государственный университет путей сообщения
(ФГБОУ ВО РГУПС)
Лискинский техникум железнодорожного транспорта имени И.В. Ковалёва
(ЛТЖТ – филиал РГУПС)

ИНФОРМАТИКА

Методическая разработка урока по теме «Представление целых и вещественных чисел в компьютере. Выполнение арифметических операций над целыми числами. Арифметические операции над числами с плавающей точкой»

для специальностей
23.02.06 Техническая эксплуатация подвижного состава железных дорог
23.02.01 Организация перевозок и управление на транспорте (по видам)

Лиски
2023

УДК 51
Конспект лекции, презентация и задания по теме Представление целых и вещественных чисел в компьютере. Выполнение арифметических операций над целыми числами. Арифметические операции над числами с плавающей точкой для студентов 1 курса очного отделения специальностей 23.02.01 Организация перевозок и управления на транспорте (по видам) (железнодорожный транспорт), 23.02.06 Техническая эксплуатация подвижного состава железных дорог и используются при проведении практической работы

Автор
Лапыгина С.Н. – преподаватель ЛТЖТ – филиала РГУПС.

Рецензент
Сергеева Т.В.– заместитель директора по УР – филиала РГУПС

Рассмотрено на заседании цикловой комиссии математических и общих естественно-научных дисциплин, протокол от 31.08.2023 №1.

Рекомендовано методическим советом ЛТЖТ – филиала РГУПС, протокол от 01.09.2023 №1.

Аннотация

Методическая разработка содержит конспект лекций, презентацию и задания для самостоятельного выполнения по теме Представление целых и вещественных чисел в компьютере. Выполнение арифметических операций над целыми числами. Арифметические операции над числами с плавающей точкой. Рекомендуется для проведения занятия по разделу Арифметические и логические основы работы компьютера в курсе изучения дисциплины Информатика.

Содержание

План урока:

Цель урока: иметь представление о форматах чисел с фиксированной и плавающей запятой (точкой), знать понятия прямого кода, обратного кода, дополнительного кода, уметь записывать целые числа (положительные и отрицательные) в прямом, обратном и дополнительном коде, иметь представление о записи в ячейке числа с плавающей точкой.

Задачи урока:

- ***образовательные****:* закрепление знаний учащихся по теме «Представление целых чисел в компьютере».
- ***развивающие:*** совершенствование умственной и познавательной деятельности учащихся, развитие мышления учащихся.
- ***воспитательные***: сознательное усвоение материала обучающимися.

Материалы и оборудование к уроку: презентация, видеоуроки**,** конспект урока, карточки с практической работой, тренировочный тест.

Тип урока: комбинированный урок объяснения нового материала и решения примеров.

Форма проведения урока: беседа, практическая работа по решению задач, парная, индивидуальная, фронтальная формы работы.

Продолжительность урока: 2 урока по 45 мин.

1. Организационный момент.
2. Фронтальный опрос по теме кодирование числовой информации:
    a. Что такое система счисления?
    b. Какие бывают системы счисления?
    c. Что называют основанием системы счисления?
    d. В каком виде находятся в памяти компьютера числовые значения?
    e. Что нужно сделать чтобы увидеть двоичный код десятичного числа?
    f. Расскажите алгоритм перевода числа из десятичной системы счисления в двоичную.
    g. Для каких систем счисления справедлив этот алгоритм?
    h. Какая система счисления называется позиционной?
    i. Приведите пример непозиционной системы счисления.
3. Практическая работа с группой по прошлой теме: 6 человек работают с тестом, остальные выполняют расчетные задания в тетради:

Расположить в порядке возрастания числа

1. 1100_2
2. F_{16}
3. 45_{10}
4. 1101_2
5. 25_{10}
6. $1B_{16}$

7. 21_8
8. 34_8

   Ответ: 1100_2, 1101_2, F_{16}, 21_8, 25_{10}, $1B_{16}$, 34_8, 45_{10}

Выполните вычисления в двоичной системе счисления:

$110111,1_2-101111,1_2+1110101_2= 1111101_2$

Выполните вычисления в восьмеричной системе счисления:

$4567_8+7134_8-1775_8= 11726_8$

Выполните вычисления и запишите результат в десятичной системе счисления:

$1001011,11_2+152,7_8 - 4D_{16}=259,5_{10}$

Восстановите неизвестные цифры в примере на сложение в восьмеричной системе, которые обозначены знаком вопроса:

3?42
215?
?721

Ответ: 3542
2157
5721

4. Объяснение нового материала.

Для хранения чисел в памяти компьютера используется два формата: *целочисленный (естественная форма)* и *с плавающей точкой* (точка — разделительный знак для целой и дробной части числа).

Целочисленный формат *(его ещё называют формат с фиксированной точкой)* используется для представления в компьютере целых *(*англ. *integer)* положительных и отрицательных чисел. Для этого, как правило, выделяются 1, 2 или 4 байта.

В форме с фиксированной запятой числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой (или точки), отделяющей целую часть от дробной.

Эта форма проста и привычна для большинства пользователей, но имеет небольшой диапазон представления чисел и поэтому не всегда пригодна при вычислениях. Если же в результате какой-либо арифметической операции получается число, выходящее за допустимый диапазон, то происходит переполнение разрядной сетки, и все дальнейшие вычисления теряют смысл.

Однобайтовое представление применяется только для положительных целых чисел. В этом формате отсутствует знаковый разряд. Наибольшее двоичное число, которое может быть записано при помощи 1 байта, равно 11111111, что в десятичной системе счисления соответствует числу 255.

Для положительных и отрицательных целых чисел обычно используется 2 и 4 байта, при этом старший бит выделяется под знак числа: 0 - плюс, 1 - минус.

Самое большое (по модулю) целое число со знаком, которое может поместиться в 2-байтовом формате, это число 0111111111111111, то есть при помощи подобного кодирования можно представить числа от −3276810 до 3276710.

Обратите внимание!

Если число вышло за указанные границы, произойдет переполнение! Поэтому при работе с большими целыми числами под них выделяется больше места, например 4 байта.

Формат с плавающей точкой используется для представления в компьютере действительных чисел (англ. real). Числа с плавающей точкой размещаются, как правило, в 4 или 8 байтах.

Нормализованная форма представления чисел с плавающей точкой обеспечивает огромный диапазон их записи и является основной в современных ЭВМ.

Представление целого положительного числа в компьютере

Рассмотрим кодирование прямым кодом:

Для представления целого числа в компьютере используется следующее правило:

- ✓ число переводится в двоичную систему; - результат дополняется нулями слева в пределах выбранного формата;
- ✓ для обозначения знака числа при любой длине ячейки памяти выделяется самый левый (самый старший) бит. Запомните: для положительных чисел знаковый бит равен 0, а для отрицательных чисел знаковый бит равен 1.

Например, при 8 битовой (однобайтовой) ячейке памяти число $\mathbf{+118_{10}}$ будет записано в двоичном коде так:

0	**1**	**1**	**1**	**0**	**1**	**1**	**0**

А число -118_{10} будет отличаться лишь первым "знаковым" битом, который на этот раз будет равен 1

1	**1**	**1**	**1**	**0**	**1**	**1**	**0**

Запомните *способ кодирования целых чисел со знаком, когда код положительного отличается от кода своего отрицательного эквивалента только знаковым битом (0 или 1) называется* ***прямым кодом****.*

Недостатки прямого кода:

1. Наличие +0 и -0. В прямом коде десятичному числу 0 соответствуют 2 кода: 00000000 и 10000000. Это неизбежно приводит к "ступору" любого электрического сумматора.

2. Операция суммирования числа со своим отрицательным эквивалентом не приводит к получению нуля. Например, просуммируем числа +5 и -5:

0	**0**	**0**	**0**	**0**	**1**	**0**	**1**
1	**0**	**0**	**0**	**0**	**1**	**0**	**1**
1	**0**	**0**	**0**	**1**	**0**	**1**	**0**

$10001010_2 = -10_{10}$

Как видите в результате суммирования получили -1010. Это является абсолютно недопустимым для технических устройств.

Обратный код для целых чисел со знаком

В обратном коде для обозначения знака также выделяется старший бит, но отрицательный эквивалент числа формируется по-другому. Для образования обратного кода отрицательного числа нужно изменить "знаковый" бит на 1 и инвертировать все биты числа (0 заменить 1, и наоборот)

Рассмотрим числа **+5** и **-5**:

+5:= $0\ 0000101_2$

-5:= $1\ 1111010_2$

Что мы имеем в результате суммирования +5 и -5?

0	**0**	**0**	**0**	**0**	**1**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Все единицы кода - это отрицательный ноль, т.к. он образовался из положительного нуля путем инвертирования всех битов

Вывод по обратному коду: Устранен один из недостатков прямого кода - при суммировании положительного и эквивалентного отрицательного числа получается ноль,

Но недостаток, связанный с наличием двух нулей не устранен: в обратном коде имеются два нуля: **00000000_2** и **11111111_2**. Следовательно, эта система представления не подходит для технических устройств.

Дополнительный код целых чисел со знаком

В дополнительном коде отрицательный эквивалент положительного числа образуется в два приема:

1. У положительного двоичного числа инвертируются биты (т.е. получаем обратный код числа)

2. К получившемуся числу прибавляется 1.

Посмотрим теперь на двоичные коды чисел +5 и -5 в дополнительном коде:

+5: 0 0000101$_2$

Получим дополнительный код числа -5:

1. Получаем обратный код от положительного числа +5 (переворачиваем биты) : 1 1111010

2. К младшему разряду получившегося двоичного числа прибавляем 1

Получим дополнительный код:

1	**1**	**1**	**1**	**1**	**0**	**1**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**	**1**	**1**

Т.е. число -5 в дополнительном коде имеет код: **1 1111011$_2$**.

Проверим получится ли ноль при суммировании +5 и -5?

0	**0**	**0**	**0**	**0**	**1**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**	**1**	**1**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Прекрасно!!! Наконец-то мы получили заветный ноль! Теперь мы знаем, что дополнительный код должен подойти для компьютерной техники.

Например, представим число **−135$_{10}$** в 2-байтовом формате:

-135$_{10}$ = 10000111 (перевод десятичного числа без знака в двоичный код);

-0000000010000111 (дополнение двоичного числа нулями слева в пределах двухбайтового формата) ;-0000000010000111 =1111111101111000 (перевод в обратный код);

-1111111101111000 – 1111111101111001 (перевод в дополнительный код).

Отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа. *Дополнительный код* позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие, позволяет упростить конструкцию арифметико-логического устройства компьютера путем

Задания для закрепления материала:

Записать внутреннее представление следующих десятичных чисел, используя 8 – разрядную ячейку: **64_{10} - 120_{10}** Прямой код Обратный код Дополнительный код

Как запишутся в оперативной памяти компьютера следующие десятичные числа в 16-ти разрядной сетке **57_{10} 200_{10} -117_{10}** Прямой код Обратный код Дополнительный код

Представление вещественного (действительного) числа в компьютере

Любое вещественное число может быть представлено в экспоненциальном виде, например:

1600000010=0,16·108

−0,000015610=−0,156·10−4

В этом формате вещественное число (R) представляется в виде произведения мантиссы (m) и основания системы счисления (P) в целой степени (n), называемой порядком.

Представим это в общем виде, как: $R=m \cdot P^n$.

Порядок n указывает, на какое количество позиций и в каком направлении должна сместиться в мантиссе точка (запятая), отделяющая дробную часть от

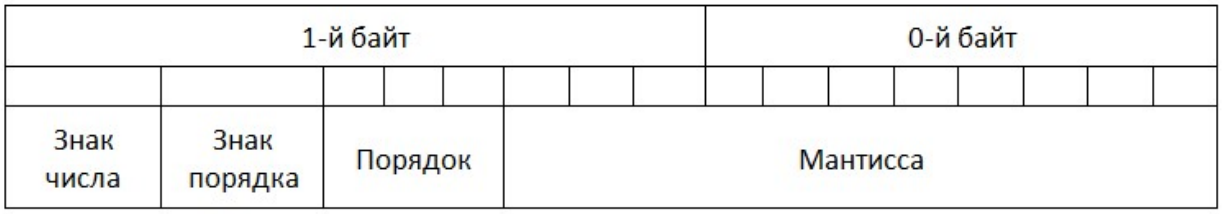
целой. Мантисса, как правило, нормализуется, то есть представляется в виде правильной дроби $0 < m < 1$.

Мантисса должна быть правильной дробью, у которой первая цифра после точки (запятой в обычной записи) отлична от нуля. Если это требование выполнено, то число называется нормализованным.

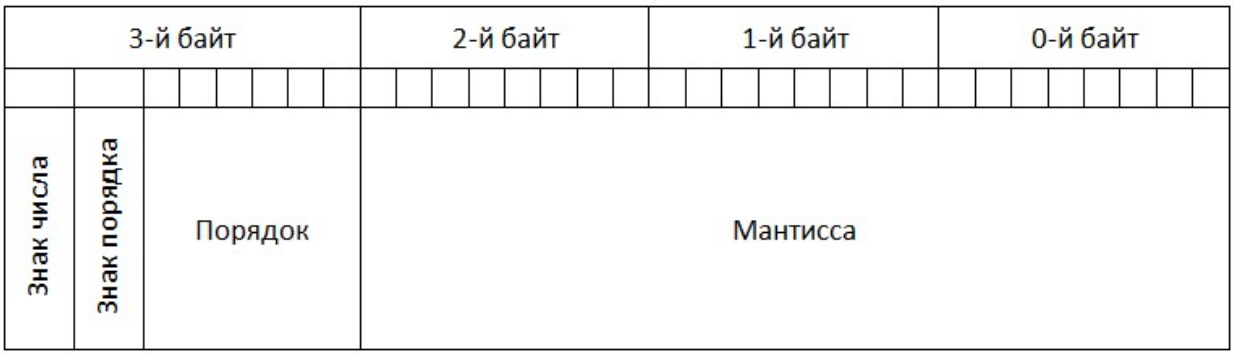
При представлении в компьютере действительного числа с плавающей точкой тоже используется нормализованная мантисса и целый порядок. И мантисса и порядок представляются в двоичном виде, как это было описано выше.

Для размещения вещественного числа обычно используется 2 или 4 байта.

В 2-байтовом формате представления вещественного числа первый байт и три разряда второго байта выделяются для размещения мантиссы, в остальных разрядах второго байта размещаются порядок числа, знаки числа и порядка.

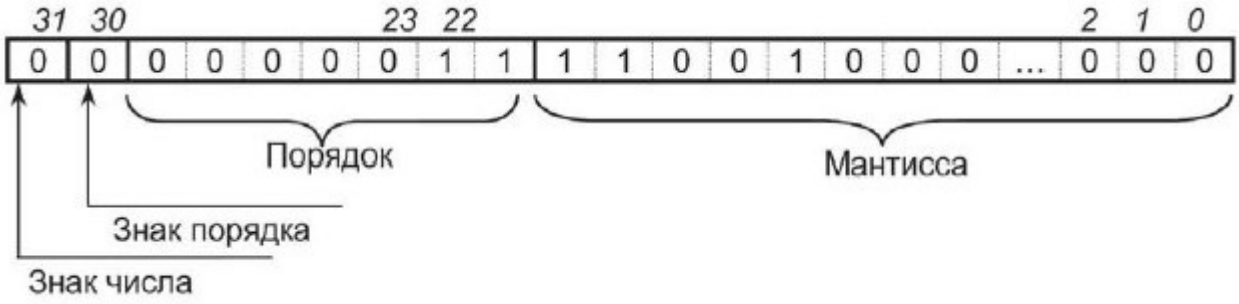


В 4-байтовом формате представления вещественного числа первые три байта выделяются для размещения мантиссы, в четвертом байте размещаются порядок числа, знаки числа и порядка.



Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.

Пример записи числа $6{,}25_{10}=110{,}01_2=0{,}11001\cdot 2^{11}$, представленного в нормализованном виде, в четырёхбайтовом формате с семью разрядами для записи порядка.



Задание 1. Запишите числа в беззнаковом коде (формат 1 байт):

а) 31; б) 163; в) 65; г) 128.

Задание 2. Найдите десятичные представления чисел, записанных в беззнаковом коде: а) 0 1011000; б) 1 0011011; в) 0 1101001; г) 1 1000000.

Задание 3. Записать число в прямом, обратном и дополнительном кодах (формат 1 байт): а) 11010; б) -11101; в) -101001; г) -1001110.

Задание 4. Запишите числа в прямом коде (формат 1 байт):

а) 31; б) -63; в) 65; г) -122.

Задание 5. Запишите числа в обратном и дополнительном кодах (формат 1 байт):

а) 9; б) -15; в) -127; г) -120.

Задание 6. Найдите десятичные представления чисел, записанных в дополнительном коде:

а) 1 1111000; б) 1 0011011; в) 1 1101001; г) 1 0000000.

Задание 7. Найдите десятичные представления чисел, записанных в обратном коде:

а) 1 1101000; б) 1 0011111; в) 1 0101011; г) 1 0000000.

Вопросы учащимся:

Назовите алгоритмы перевода чисел в обратный и дополнительный коды:

Обратный код.

Записать двоичный код абсолютной величины числа.

Инвертировать все цифры двоичного кода абсолютной величины числа (модуля числа), включая разряд знака: нули заменяются единицами, а единицы — нулями.

Дополнительный код отрицательного числа.

Модуль числа записать в прямом коде в n двоичных разрядах. (Для этого получить внутреннее представление положительного числа N: перевести число N в двоичную систему счисления, полученный результат дополнить слева незначащими нулями до k разрядов)

Получить обратный код числа, для этого значения всех битов инвертировать (все единицы заменить на нули и все нули заменить на единицы).

К полученному обратному коду прибавить единицу.

В чем вы видите достоинства представления чисел в формате с фиксированной запятой?

Ответ: простота и наглядность представления чисел, простота алгоритмов реализации арифметических операций.

Рассмотрите пример записи дополнительного кода отрицательного числа -2002 для 16 разрядного компьютерного представления

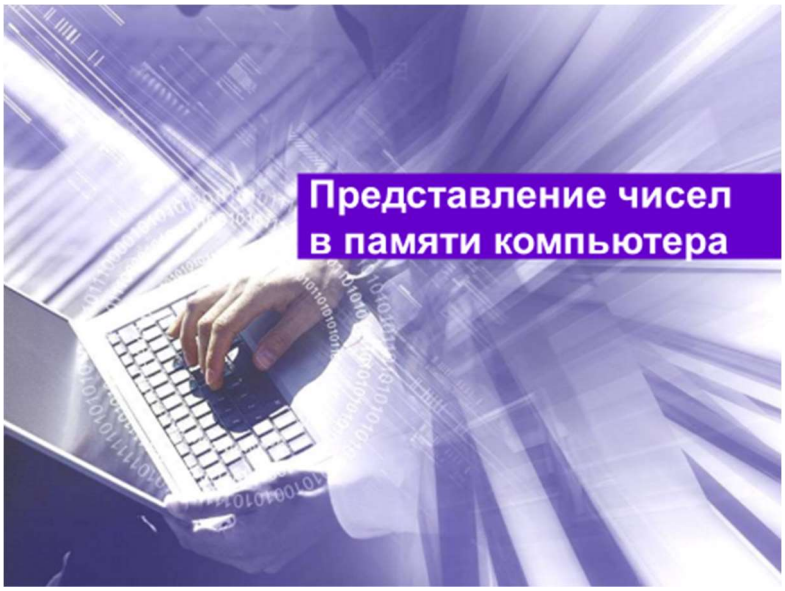
Домашняя работа

Задание 1. Получить внутреннее представление целого числа 123_{10} в 8-разрядной ячейке памяти компьютера.

Задание 2. Получить внутреннее представление целого числа - 123_{10} в 8-разрядной ячейке памяти компьютера.

Задание 3. Получить внутреннее представление целого числа - 17_{10} в 16 – ти разрядной ячейке памяти компьютера.

Слайды презентации



1. Что такое система счисления?
2. Какие бывают системы счисления?
3. Что называют основанием системы счисления?
4. В каком виде находятся в памяти компьютера числовые значения?
5. Что нужно сделать чтобы увидеть двоичный код десятичного числа?
6. Расскажите алгоритм перевода числа из десятичной системы счисления в двоичную.
7. Для каких систем счисления справедлив этот алгоритм?
8. Какая система счисления называется позиционной?
9. Приведите пример непозиционной системы счисления.

Расположите в порядке возрастания числа

1. 1100_2
2. F_{16}
3. 45_{10}
4. 110_{12}
5. 25_{10}
6. $1B_{16}$
7. 21_8
8. 34_8

Ответ: 1100_2, 1101_2, F_{16}, 21_8, 25_{10}, $1B_{16}$, 34_8, 45_{10}

Выполните вычисления в двоичной системе счисления:

$110111,1_2-101111,1_2+1110101_2 = 1111101_2$

Выполните вычисления в восьмеричной системе счисления:

$45678+71348-17758 = 11726_8$

Выполните вычисления и запишите результат в десятичной системе счисления:

$1001011,11_2+152,7_8 - 4D_{16} = 259,5_{10}$

Восстановите неизвестные цифры в примере на сложение в восьмеричной системе, которые обозначены знаком вопроса:

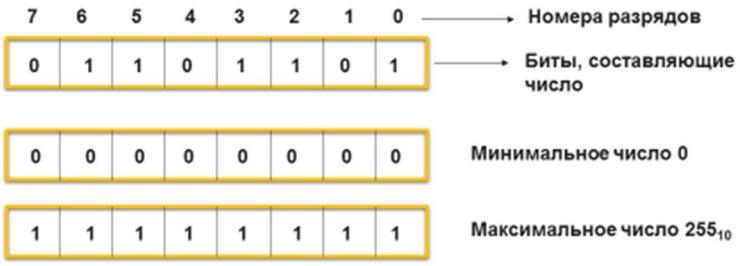
$$\begin{array}{r} 3542 \\ +\ 21?7 \\ \hline 57?1 \end{array}$$

Числовые величины

- **Целые** (формат с фиксированной запятой)
- **Вещественные** (формат с плавающей запятой)

Целые числа без знака

Для хранения *целых неотрицательных чисел без знака* обычно отводится одна ячейка памяти (8 битов).



$11111111_2 = 100000000_2 - 1 = 2^8 - 1 = 255_{10}$

Для n-разрядного представления максимальное целое неотрицательное число равно $2^n - 1$.

Целые числа без знака

Пример. Представить число 72_{10} в двоичном виде в восьмибитовом представлении в формате целого без знака.

Решение.

$72_{10} = 1001000_2$

0	1	0	0	1	0	0	0

Целые числа со знаком

Для хранения **целых чисел со знаком** отводится две ячейки памяти (16 битов).

Старший разряд числа определяет его знак.
Если он равен 0, число положительное,
если 1, то отрицательное.

$72_{10} = 1001000_2$

0	1	0	0	1	0	0	0

$-72_{10} = -11001000_2$

1	1	0	0	1	0	0	0

Такое представление чисел в компьютере называется **прямым кодом**.

Целые числа со знаком

Для n-разрядного представления со знаком (с учетом выделения одного разряда на знак):

- минимальное отрицательное число равно -2^{n-1}
- максимальное положительное число равно $2^{n-1} - 1$,

Целые числа со знаком

Для представления отрицательных целых чисел используется **дополнительный код.**

Алгоритм получения дополнительного кода отрицательного числа:

1. Число записать прямым кодом в n двоичных разрядах.
2. Получить обратный код числа, для этого значения всех битов инвертировать, кроме старшего разряда.
3. К полученному обратному коду прибавить единицу.

Представить число -2014_{10} в двоичном виде в шестнадцатибитном представлении в формате целого со знаком.

Прямой код	-2014_{10}	$10000111\ 11011110_2$
Обратный код	Инвертирование	$11111000\ 00100001_2$
	Прибавление единицы	$11111000\ 00100001_2$ $00000000\ 00000001_2$
Дополнительный код		$11111000\ 00100010_2$

Целые числа со знаком

Алгебраическое сложение двоичных чисел

1. Положительные слагаемые представить в прямом коде.
2. Отрицательные слагаемые – в дополнительном.
3. Найти сумму кодов, включая знаковые разряды, которые при этом рассматриваются как старшие разряды. При переносе из знакового разряда единицу переноса отбрасывают.
4. В результате получают алгебраическую сумму в прямом коде, если эта сумма положительная, и в дополнительном, если сумма отрицательная.

Пример 1. Найти разность $13_{10} - 12_{10}$ в восьмибитном представлении.

	13_{10}	-12_{10}
Прямой код	00001101	10001100
Обратный код	-	11110011
Дополнительный код	-	11110100

```
  0 0001101
+ 1 1110100
-----------
10 0000001
```

Так как произошел перенос из знакового разряда, первую единицу отбрасываем, и в результате получаем 00000001.

Пример 2. Найти разность $8_{10} - 13_{10}$ в восьмибитном представлении.

	8_{10}	-13_{10}
Прямой код	00001000	10001101
Обратный код	-	11110010
Дополнительный код	-	11110011

```
  0 0001000
+ 1 1110011
-----------
  1 1111011
```

Пример 2. Найти разность $8_{10} - 13_{10}$ в восьмибитном представлении.

```
  0 0001000
+ 1 1110011
-----------
  1 1111011
```

В знаковом разряде стоит 1, значит результат получен в дополнительном коде. Прейдем от дополнительного кода к обратному, вычтя единицу:

```
  1 1110011
- 0 0000001
-----------
  1 1111010
```

Прейдем от обратного кода к прямому, инвертируя все цифры, за исключением знакового (старшего) разряда: $10000101_2 = 5_{10}$.

Вещественные числа

Вещественные числа хранятся и обрабатываются в компьютере в формате *с плавающей запятой*, использующем экспоненциальную форму записи чисел.

$$A = M \times q^n$$

M – мантисса числа (правильная отличная от нуля дробь),

q – основание системы счисления,

n – порядок числа.

Диапазон ограничен максимальными значениями M и n.

Вещественные числа

Например, $123{,}45 = 0{,}12345 \cdot 10^3$

Порядок указывает, на какое количество позиций и в каком направлении должна сместиться десятичная запятая в мантиссе.

Число в формате с плавающей запятой может занимать в памяти 4 байта *(обычная точность)* или 8 байтов *(двойная точность)*.

При записи числа выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.

Мантисса **M** и порядок **n** определяют диапазон изменения чисел и их точность.

Вещественные числа

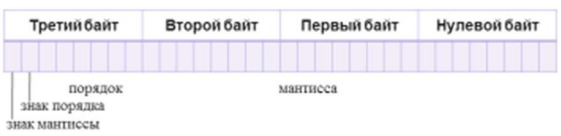
Старший байт (третий) в 4-х байтном формате включает в себя:

один бит (старший) - знак числа;

один бит - знак порядка;

шесть битов - порядок числа.

три байта (нулевой, первый и второй) хранят мантиссу числа



В таком представлении максимальный порядок числа равен $111111_2 = 63_{10}$. Следовательно, 10^{63} - максимальное число, которое можно закодировать таким образом.

Вещественные числа

Положительные и отрицательные значения порядка существенно усложняют обработку вещественных чисел. Поэтому во многих современных компьютерах используют не прямое значение порядка, а смещенное. Запись вещественного числа имеет структуру следующего вида:

Здесь порядок n-разрядного нормализованного числа задается в смещенной форме. Связь между смещенным порядком S и математическим P в данном случае выражается формулой:

$S = P + 64_{10} = P + 100\ 0000_2$.

Вещественные числа

Пример 3. Записать внутреннее представление числа 250,1875 в форме с плавающей точкой в 4-х байтовом машинном слове.

Решение:

1. Переведем число в двоичную систему счисления с 24 значащими цифрами (3 байта под мантиссу):

$250.1875_{10} = 11111010{,}0011000000000000_2$.

2. Запишем в форме нормализованного двоичного числа с плавающей точкой: $0{,}11111010001100000000000 \cdot 10^{1000_2}$. **Здесь мантисса, основание системы счисления ($2_{10} = 10_2$) и порядок ($8_{10} = 1000_2$) записаны в двоичной системе.**

3. Вычислим смещенный порядок: $S_2 = 1000 + 1000000 = 1001000$.

4. Запишем представление числа в 4-байтовой ячейке памяти с учетом знака числа:

0	1	0	0	1	0	0	0	1	1	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Домашнее задание

- Задание 1. Получить внутреннее представление целого числа 123_{10} в 8-разрядной ячейке памяти компьютера.
- Задание 2. Получить внутреннее представление целого числа - 123_{10} в 8-разрядной ячейке памяти компьютера.
- Задание 3. Получить внутреннее представление целого числа - 17_{10} в 16 – ти разрядной ячейке памяти компьютера.

Слайд 1

Список рекомендуемых источников

1 Новожилов, О. П. Информатика : учебник для СПО / О. П. Новожилов. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2021. — 620 с. — (Профессиональное образование).

2 *Гаврилов, М. В.* Информатика и информационные технологии : учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М. : Издательство Юрайт, 2021. — 383 с. — (Профессиональное образование).

3 *Зимин, В. П.* Информатика. Лабораторный практикум в 2 ч. Часть 1 : учебное пособие для СПО / В. П. Зимин. — М. : Издательство Юрайт, 2021. — 110 с. — (Профессиональное образование).

4 *Зимин, В. П.* Информатика. Лабораторный практикум в 2 ч. Часть 2 : учебное пособие для СПО / В. П. Зимин. — М. : Издательство Юрайт, 2021. — 145 с. — (Профессиональное образование).

5 Информационные технологии: Сети и телекоммуникации : учебник и практикум для академического бакалавриата / К. Е. Самуйлов [и др.] ; под ред. К. Е. Самуйлова, И. А. Шалимова, Д. С. Кулябова. — М. : Издательство Юрайт, 2017. — 363 с. — (Бакалавр. Академический курс).

6 *Сидорова, А. А.* Электронное правительство : учебник и практикум для бакалавриата и магистратуры / А. А. Сидорова. — М. : Издательство Юрайт, 2017. — 165 с. — (Бакалавр и магистр. Академический курс).

7 *Черпаков, И. В.* Основы программирования : учебник и практикум для СПО / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 219 с. — (Профессиональное образование).

8 *Черпаков, И. В.* Теоретические основы информатики : учебник и практикум для академического бакалавриата / И. В. Черпаков. — М. : Издательство Юрайт, 2017. — 353 с. — (Бакалавр. Академический курс).