Практическая работа №3
«Выполнение арифметических операций над двоичными числами»

1. Цель работы:

- Научиться конвертировать числа между десятичной, двоичной и шестнадцатеричной системами счисления, использовать двоичные числа для арифметических операций и анализировать результаты с учетом разрядности и кодирования со знаком
- Освоить методы выполнения арифметических операций над двоичными числами в электронных таблицах Excel.

2. Теоретический блок:

2.1. Системы счисления:

1) **Десятичная система (основание 10).** Используется повсеместно для представления чисел в обычной жизни. В этой системе используются цифры от 0 до 9. Примеры перевода из десятичной системы в двоичную и шестнадцатеричную системы схожи – необходимо поделить число десятичной системы на новое основание. Для двоичной системы необходимо последовательное деление на 2, с записью остатков в обратном порядке:

$$93_{10} = 1011101_2$$

Это получилось в результате деления с записью остатка от каждой операции в обратном порядке:

- 93 ÷ 2 = 46, остаток 1
- 46 ÷ 2 = 23, остаток 0
- 23 ÷ 2 = 11, остаток 1
- 11 ÷ 2 = 5, остаток 1
- 5 ÷ 2 = 2, остаток 1
- 2 ÷ 2 = 1, остаток 0
- 1 ÷ 2 = 0, остаток 1

В случае с переводом числа из десятичной в шестнадцатеричную систему необходимо поделить число на 16 и записывать остаток, в обратном порядке.

Для числа 43981:

$$43981_{10} = ABCD_{16}$$

Это получилось в результате:

- 43981÷16=2748, остаток 13 (D в шестнадцатеричной системе)
- 2748÷16=171, остаток 12 (C в шестнадцатеричной системе)
- 171÷16=10, остаток 11 (B в шестнадцатеричной системе)
- 10÷16=0, остаток 10 (A в шестнадцатеричной системе)

2) **Двоичная система (основание 2).** Используется в компьютерах и цифровой технике. В этой системе используются только цифры 0 и 1. В двоичной системе каждое число представляется как последовательность цифр, где каждая цифра (или бит) соответствует определенной степени двойки. Возьмем двоичное число 1011_2. Чтобы понять, какое десятичное число оно представляет, разложим его на составляющие:

$$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

Для перевода двоичного числа в шестнадцатеричную систему счисления используется следующий алгоритм:

1. **Разделение числа на группы по 4 бита**. Начиная с младших разрядов (справа) и разделяя двоичное число на группы по 4 бита. Если количество битов не кратно 4, добавляем ведущие нули слева.
2. **Преобразование каждой группы в соответствующий шестнадцатеричный символ**. Каждая 4-битная группа преобразуется в соответствующий символ шестнадцатеричной системы (от 0 до F).
3. **Запись полученных символов слева направо**. Полученные шестнадцатеричные символы записываются слева направо, начиная с самой левой группы.

Возьмем двоичное число 1011_2. Число уже содержит 4 бита, дополнительных нулей не требуется. Число является 11 в 10-чной системе, таким образом, $1011_2 = B_{16}$.

Возьмем более сложный пример. Переведем двоичное число 110101101011_2 в 16-чную систему. Разделим число по 4 бита: 1101 0110 1011. Далее каждая группа отдельно переводится в 16-чную систему. Так 1101_2 = D_{16} (где D_{16} = 13_{10}), 0110_2 = 6_{16} (где 6_{16} = 6_{10}), 1011_2 = B_{16} (где B_{16} = 11_{10}). Таким образом:

$$110101101011_2 = D6B_{16}$$

Пример с добавлением ведущих нулей. Возьмем число 101_2 и переведем в 16-чную систему. Число содержит 3 бита из 4х, необходимо добавить ноль в начало числа, таким образом – 101_2 = 0101_2. Далее все происходит по схожему алгоритму, что и в предыдущих алгоритмах. 0101_2 = 5_{10} = 5_{16}.

3) **Шестнадцатеричная система (основание 16).** В этой системе используются цифры от 0 до 9 и буквы A, B, C, D, E, F для представления чисел 10-15. Для перевода шестнадцатеричного числа в десятичную систему необходимо разложить число на суммы произведений его цифр на соответствующие степени числа 16. Рассмотрим несколько примеров:

1. Переведем число $2A3_{16}$ в десятичную систему. Разложим число на составляющие:
   $2A3_{16} = 2 \times 16^2 + A \times 16^1 + 3 \times 16^0$,
   Здесь:
   - $2*16^2 = 512$,
   - A представляет собой 10, значит $A*16^1 = 10*16 = 160$,
   - $3*16^0 = 3*1 = 3$.

Таким образом, $2A3_{16} = 512+160+3 = 675_{10}$.

2. Переведем число $B9_{16}$ в двоичную систему.
   - $B_{16} = 1011_2$
   - $9_{16} = 1001_2$

   $B9_{16} = 10111001$

В данном случае значения замен зависят от прямого соответствия между 16-чными цифрами и их двоичными коэффициентами. Сравните самостоятельно по таблице ниже:

Шестнадцатеричная цифра	Десятичное значение	Двоичное значение (4 бита)
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
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Перевод чисел с дробной частью между системами счисления

Вспомним также как и работать с числами, у которых есть дробная часть. Перевод таких чисел состоит из двух алгоритмов:

- **Алгоритм с делением для целой части числа.**
- **Алгоритм с умножением для дробной части числа.**

Алгоритм с делением для целой части числа был описан ранее. Необходимо разделить число на основание новой системы счисления, записать остаток и повторить пока частное не станет равно 0. Запись остатков происходит в обратном порядке.

Для дробной части числа же используется другой алгоритм. Необходимо умножить дробную часть на основание новой системы счисления, записав целую часть результата умножения. Повторяем операцию с дробной частью полученного числа до тех пор, пока не получится целое число, либо до необходимого количества знаков после запятой. Однако этот алгоритм изменяется в зависимости от систем счисления. Рассмотрим 2 примера:

- Перевод 45.625_{10} в шестнадцатеричную систему. Делим число на целую и дробную части. Целая часть 45_{10} = $2D_{16}$. Дробная часть 0.625 умножается на 16, что дает 10 (или A в 16-ной системе). То есть $0.625_{10} = 0.A_{16}$. Результат: $45.625_{10} = 2D.A_{16.}$
- Перевод 1011.11_2 в десятичную систему. Как и в предыдущем примере, число делится на целую и дробную части. По правилам, описанным ранее, установим, что $1011_2 = 11_{10}$. Этот ответ получен в результате умножения каждой цифры на основание системы 2. Для дробной части действуют те же правила: цифры умножаются на основание системы в степени, которая соответствует их позиции, но со знаком минус (начиная с 1). Так для дробной части $11_2 = 1*2^{-1}+1*2^{-2} = 0.5+0.25 = 0.75$. Результат: $1011.11_2 = 11.75_{10}$.

2.2. Арифметические операции в двоичной системе

- **Сложение и вычитание**. В двоичной системе сложение и вычитание выполняются по тем же правилам, что и в десятичной, но с учётом двоичной логики (например, $1 + 1 = 10_2$).

Основное правило заключается в следующем:

- $0 + 0 = 0$

- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$ (что эквивалентно 2 в десятичной системе, но записывается как 10 в двоичной).

Пример сложения:

$$
\begin{array}{r}
1011_2\ (11_{10}) \\
+ \\
1101_2\ (13_{10}) \\
\text{--------} \\
11000_2\ (24_{10})
\end{array}
$$

В этом примере 1 + 1 даёт 10, записываем 0 и переносим 1 в следующий разряд.

Пример вычитания:

$$
\begin{array}{r}
1101_2\ (13_{10}) \\
-\ 1011_2\ (11_{10}) \\
\text{--------} \\
0010_2\ (2_{10})
\end{array}
$$

Здесь, если встречается ситуация, когда нужно вычесть 1 из 0, необходимо "занять" 1 у следующего старшего разряда.

Дополнительный код. Дополнительный код используется для представления отрицательных чисел в двоичной системе, особенно в компьютерах. Этот метод позволяет легко выполнять арифметические операции с отрицательными числами. Преобразование числа в дополнительный код включает в себя следующие шаги:

1. Записываем число в прямом коде (обычная двоичная запись).
2. Инвертируем все биты (0 превращаем в 1, а 1 в 0).
3. Добавляем 1 к полученному значению.

Пример для числа -5 в 3-битном формате:

1. Прямой код для 5: 101_2
2. Инвертируем биты: 010_2
3. Добавляем 1: 011_2

Однако, при использовании небольшого количества битов, таких как 3 бита, могут возникнуть ограничения на диапазон представляемых чисел.

2.3. Применение 16-разрядного двоичного формата со знаком

Формат данных

В 16-разрядном формате со знаком числа представляют с использованием 16 бит. Этот формат включает:

1. **Знаковый бит**. Первый бит (самый старший бит) используется для обозначения знака числа:
   - 0 — положительное число.
   - 1 — отрицательное число.
2. **Числовое значение**. Остальные 15 битов используются для представления значения числа.

Этот формат позволяет представить как положительные, так и отрицательные числа, где отрицательные числа представлены в дополнительном коде.

Примеры

1. **Положительные числа**:

   - Число 5 в 16-разрядном формате со знаком: 0000 0000 0000 0101.
   - Число -5 в 16-разрядном формате со знаком (дополнительный код): 1111 1111 1111 1011.

2. **Отрицательные числа**:

   - Для получения представления отрицательного числа, сначала нужно найти его абсолютное значение в двоичном формате, затем инвертировать биты и прибавить 1. Например, для -5:

1. **Абсолютное значение 5**: 0000 0000 0000 0101
2. **Инвертирование битов**: 1111 1111 1111 1010
3. **Прибавление 1**: 1111 1111 1111 1011

Диапазон значений

В 16-разрядном формате со знаком диапазон значений определяется следующим образом:

1. **Положительные числа**: Начинаются с 0000 0000 0000 0000 (0) до 0111 1111 1111 1111 (32767). Это соответствует диапазону от 0 до 32767.

2. **Отрицательные числа**: Начинаются с 1000 0000 0000 0000 (-32768) до 1111 1111 1111 1111 (-1). Это соответствует диапазону от -32768 до -1.

Итак, в 16-разрядном формате со знаком:

- Минимальное значение: -32768
- Максимальное значение: 32767

Примеры

1. **Положительное число**: 12345 в десятичной системе. В двоичном формате: 0011 0000 0011 1001.

2. **Отрицательное число**: -12345 в десятичной системе. В двоичном формате:

   - Абсолютное значение 12345 в двоичной системе: 0011 0000 0011 1001
   - Инвертирование битов: 1100 1111 1100 0110
   - Прибавление 1: 1100 1111 1100 0111

Таким образом, представление -12345 в 16-разрядном формате со знаком: 1100 1111 1100 0111.

Этот формат удобен для работы с вычислениями, поскольку позволяет легко выполнять арифметические операции и эффективно использовать ресурсы памяти.

2.4. Анализ флагов состояния процессора

Флаги состояния процессора — это специальные биты, которые хранятся в регистре флагов (также известном как регистр состояния) центрального процессора (ЦП). Эти флаги отражают результаты различных арифметических и логических операций, выполняемых процессором. Они важны для принятия решений при выполнении программ, таких как условные переходы, обработка исключений и другие действия.

Основные флаги состояния

1. **Флаг переноса (Carry Flag, CF)**
   - Устанавливается в случае переноса из старшего бита при выполнении операции сложения или заимствования при вычитании.
   - **Пример -** при сложении двух чисел, если результат операции требует большего количества битов, чем доступно, устанавливается флаг переноса.
   - **Пример сложения**:
     - При сложении 32767 и 1 (в 16-битной системе):
       - 32767 (0000 0111 1111 1111)
       - 1 (0000 0000 0000 0001)
       - Результат сложения: 32768 (1000 0000 0000 0000), который выходит за пределы 16-битного диапазона.
       - Устанавливается флаг переноса (CF = 1).
2. **Флаг нуля (Zero Flag, ZF)**

- Устанавливается, если результат арифметической операции равен нулю.
- **Пример** - при вычитании 5 из 5:
  - 5 (0000 0000 0000 0101)
  - -5 (1111 1111 1111 1011, в дополнительном коде)
  - Результат вычитания: 0 (0000 0000 0000 0000).
  - Устанавливается флаг нуля (ZF = 1).

3. **Флаг знака (Sign Flag, SF)**

   - Устанавливается в соответствии со знаком результата операции. Если старший бит результата равен 1, флаг знака устанавливается (для отрицательного числа), если 0 — сбрасывается (для положительного числа).
   - **Пример** - при выполнении операции -3 + 2:
     - -3 (1111 1111 1111 1101)
     - 2 (0000 0000 0000 0010)
     - Результат операции: -1 (1111 1111 1111 1111).
     - Устанавливается флаг знака (SF = 1).

4. **Флаг переполнения (Overflow Flag, OF)**

   - Устанавливается при переполнении результата арифметической операции, когда результат выходит за пределы диапазона, который можно представить с заданным числом битов (например, 16 бит).
   - **Пример -** при сложении двух положительных чисел, результат которых превышает максимальное значение, которое можно представить:
     - При сложении 30000 и 10000:
       - 30000 (0111 0101 1000 0000)
       - 10000 (0010 0110 1000 0000)
       - Результат сложения: 40000 (в 16-битной системе: 0010 0110 0100 0000 0000), что превышает 16-битный диапазон.
       - Устанавливается флаг переполнения (OF = 1).

Дополнительные флаги

1. **Флаг четности (Parity Flag, PF)**

   - Устанавливается, если количество единичных битов в результатах операции четное.
   - **Пример** – при выполнении операции с числом 5 (0000 0000 0000 0101):
     - Количество единиц: 2 (четное).
     - Устанавливается флаг четности (PF = 1).

2. **Флаг вспомогательного переноса (Auxiliary Carry Flag, AF)**

- Устанавливается при переносе из младшего полубайта (из 4 младших битов) при сложении или заимствовании при вычитании.
- **Пример** – при сложении 15 и 1 (в 4-битной системе):
  - 15 (1111)
  - 1 (0001)
  - Результат сложения: 16 (10000), перенос из младшего полубайта.
  - Устанавливается флаг вспомогательного переноса (AF = 1).

Пример работы с флагами

Рассмотрим пример сложения двух 16-битных чисел:

- 32767 (0000 0111 1111 1111)
- 1 (0000 0000 0000 0001)

Результат сложения:

- 32768 (1000 0000 0000 0000)

Флаги состояния:

- **Флаг переноса (CF)** - установлен (1), так как результат превышает максимальное значение 16-битного диапазона.
- **Флаг нуля (ZF)** - сброшен (0), так как результат не равен нулю.
- **Флаг знака (SF)** - установлен (1), так как старший бит результата равен 1.
- **Флаг переполнения (OF)** - установлен (1), так как произошло переполнение.
- **Флаг четности (PF)** - зависит от количества единиц в результате (в данном случае 1000 0000 0000 0000 — 1 единица, нечетное количество, PF = 0).
- **Флаг вспомогательного переноса (AF)** - не установлен в данном примере.

3. Практический блок

Для практической работы будет использоваться MS Excel. Excel имеет большое количество инструментов и функций, для решения задач, связанных с двоичной, десятичной и шестнадцатеричной системами счисления, а также для выполнения арифметических операций и анализа флагов:

1. DEC2BIN/ДЕС.В.ДВ (Преобразует десятичное число в двоичное)
2. BIN2DEC/ДВ.В.ДЕС (Преобразует двоичное число в десятичное)
3. DEC2HEX/ДЕС.В.ШЕСТН (Преобразует десятичное число в шестнадцатеричное)
4. HEX2DEC/ШЕСТН.В.ДЕС (Преобразует шестнадцатеричное число в десятичное)
5. HEX2BIN/ШЕСТН.В.ДВ (Преобразует шестнадцатеричное число в двоичное)
6. BIN2HEX/ДВ.В.ШЕСТН (Преобразует двоичное число в шестнадцатеричное)
7. IF/ЕСЛИ (Проверяет условие и возвращает одно значение, если условие истинно, и другое — если ложно)
8. SUM/СУММ (Складывает значения в указанных ячейках)
9. SUBTRACT/- (Вычитает одно число из другого)

Разберем несколько примеров с применением данных функций Excel.

3.1. Перевод чисел

Переведите десятичное число «57» в двоичную и шестнадцатеричную системы счисления.

Решение:

Для перевода в двоичную систему используйте формулу = ДЕС.В.ДВ(57). Результат: 111001.

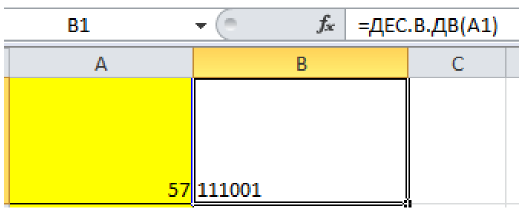

Рисунок 1 – перевод в двоичную систему из десятичной

Для перевода в шестнадцатеричную систему используйте формулу = ДЕС.В.ШЕСТН(57). Результат: 39.

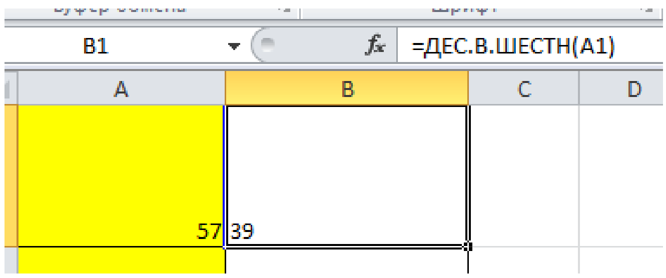

Рисунок 2 – перевод в шестнадцатеричную систему из десятичной

Переведите двоичное число 110011 в десятичную и шестнадцатеричную системы.

Решение:

- В десятичную систему:
  - Используйте формулу = ДВ.В.ДЕС(110011). Результат: 51.

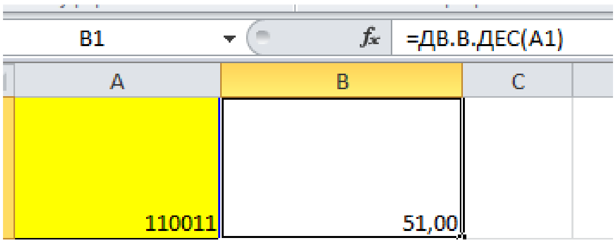

Рисунок 3 – перевод в десятичную систему из двоичной

- В шестнадцатеричную систему:
  - Сначала переводим двоичное число в десятичное 51 с помощью ДВ.В.ДЕС.
  - Затем используем формулу = ДЕС.В.ШЕСТН(51). Результат: 33.

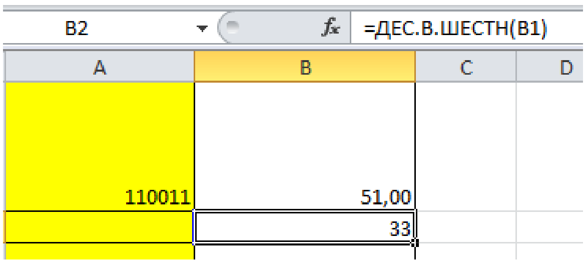

Рисунок 4 – Перевод в шестнадцатеричную систему из двоичной

3.2. Арифметические операции

Выполните сложение двоичных чисел 1101 и 1011 в Excel.

Решение:

- Сначала переведите двоичные числа в десятичные:
  - = ДВ.В.ДЕС(1101) вернет 13.
  - = ДВ.В.ДЕС (1011) вернет 11.
- Выполните сложение 13 + 11 = 24.
- Переведите результат обратно в двоичную систему:
  - = ДЕС.В.ДВ(24) вернет 11000.

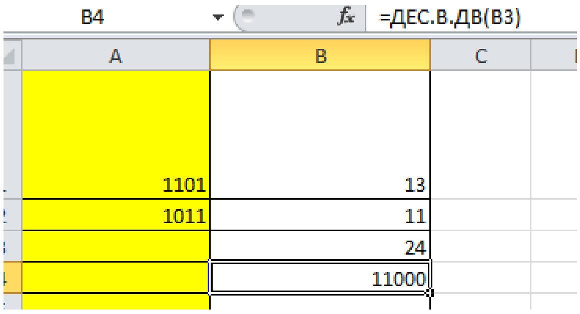

Рисунок 5 – сложение двоичных чисел

Вычтите число 1001 из 1101, используя двоичный формат.

Решение:

- Переведите оба числа в десятичные:
    - =ДВ.В.ДЕС (1101) вернет 13.
    - =ДВ.В.ДЕС (1001) вернет 9.
- Выполните вычитание 13 - 9 = 4.
- Переведите результат обратно в двоичную систему:
    - =ДЕС.В.ДВ(4) вернет 100.

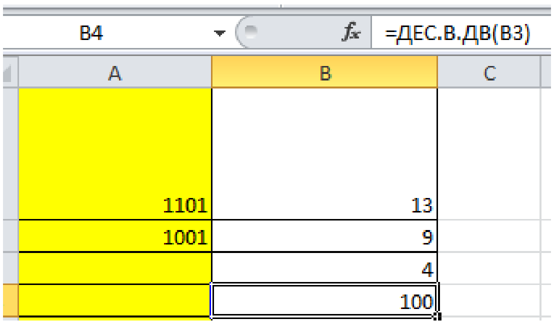

Рисунок 6 – вычитание двоичных чисел

3.3. Работа с флагами

3.3.1. Выполните сложение чисел 32767 и 1, и определите, установлен ли флаг переполнения.

Решение:

- Выполните сложение =32767 + 1.
- Для определения переполнения в 16-разрядном формате, если результат больше 32767 (максимально возможное положительное число для 16 бит), это означает переполнение:
    - Используйте формулу: =IF(32767 + 1 > 32767, "Переполнение", "Нет переполнения").
    - Результат: Переполнение.

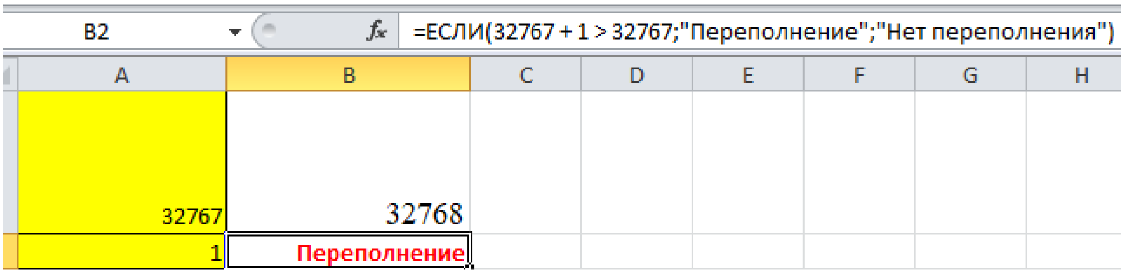

Рисунок 7 – условие для флага переполнения

3.3.2. Проверьте флаг переноса при сложении чисел 255 и 1 (в 8-разрядном формате).

Решение:

- Выполните сложение =255 + 1.
- В 8-разрядном формате, если результат превышает 255 (максимальное значение для 8 бит), происходит перенос:
  - Используйте формулу: =IF(255 + 1 > 255, "Перенос", "Нет переноса").
  - Результат: Перенос.

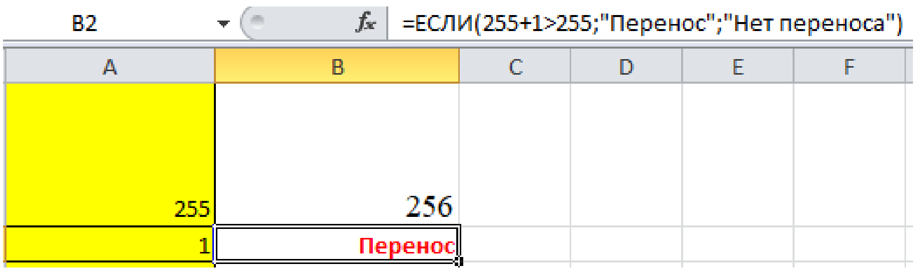

Рисунок 8 – условие для флага переноса

4. Задание для отчета по лабораторной работе

- Определите два числа A и B, соответствующие вашему варианту
- Выполните сложение (или вычитание) чисел A и B в 16-разрядном двоичном формате со знаком в Excel
- После выполнения операции проанализируйте установленные флаги состояния (перенос, переполнение, знак, ноль)
- Переведите результат обратно в десятичную систему счисления и прокомментируйте полученные значения.
- Оформите отчет

5. Варианты для самостоятельной работы

1) A = 12345, операция: сложение, B = 23456
2) A = 10000, операция: вычитание, B = 30000
3) A = 32767, операция: сложение, B = 1 (проверка переполнения)
4) A = -16384, операция: сложение, B = -16384 (проверка переноса)
5) A = 21845, операция: вычитание, B = 32767
6) A = -32768, операция: сложение, B = 32767
7) A = 16383, операция: вычитание, B = 32767
8) A = 30500, операция: сложение, B = 2222
9) A = 32760, операция: сложение, B = 10
10) A = 5000, операция: вычитание, B = 3000
11) A = -12345, операция: сложение, B = -10000
12) A = 20000, операция: сложение, B = 15000
13) A = -32767, операция: сложение, B = 1
14) A = 11111, операция: сложение, B = 22222
15) A = -12345, операция: вычитание, B = 12345
16) A = 24576, операция: сложение, B = 8192
17) A = 32766, операция: сложение, B = 2 (проверка переполнения)
18) A = -20000, операция: сложение, B = 32767
19) A = 32760, операция: вычитание, B = 32760
20) A = 15000, операция: сложение, B = 17767
21) A = -32768, операция: сложение, B = -32768 (проверка переноса)
22) A = 12345, операция: вычитание, B = 54321
23) A = 32700, операция: сложение, B = 100
24) A = -25000, операция: сложение, B = -7777
25) A = 32750, операция: сложение, B = 10
26) A = -32760, операция: сложение, B = 32760
27) A = 32767, операция: вычитание, B = 32767
28) A = 16000, операция: сложение, B = 16000
29) A = -10000, операция: сложение, B = 32767
30) A = 10000, операция: сложение, B = 22768

6. Вопросы для самостоятельного контроля

1. Как перевести число из двоичной системы в шестнадцатеричную систему?
2. Что такое дополнительный код?
3. Как определить знак числа?
4. В каких случаях флаги состояний процессора CF и OF будут совпадать?
5. В каких случаях флаг состояния ZF будет равен 1?

7. Список литературы для самостоятельного изучения

1. Хеннесси, Дж. Л., Паттерсон, Д. А. Компьютерная архитектура: количественный подход. — 5-е изд. — М.: Вильямс, 2016. — 944 с.
2. Таненбаум, Э. Архитектура компьютера. Структурный подход. — 5-е изд. — СПб.: Питер, 2013. — 832 с.
3. Архитектура вычислительных систем [Электронный ресурс]: учебное пособие – Эл. изд. - Электрон. текстовые дан. (1 файл pdf: 77 с.). - Грейбо С.В., Новосёлова Т.Е., Пронькин Н.Н., Семёнычева И.Ф. 2019