Лабораторная работа № 3

**Лабораторная работа 1**

**Арифметические и логические основы ЭВМ**

**1 СИСТЕМЫ СЧИСЛЕНИЯ И ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В**

## ЭВМ

##### 1.1 ПОНЯТИЕ ОБ ОСНОВНЫХ СИСТЕМАХ СЧИСЛЕНИЯ

Под **системой счисления** понимается способ представления любого числа с помощью некоторого алфавита символов, называемых цифрами. Все системы счисления делятся на **позиционные** и **непозиционные**.

**Непозиционными** называются такие системы счисления, в которых каждый символ сохраняет свое значение независимо от места его положения в числе. Примером непозиционной системы счисления является римская система, в которой символам I, V, X, L, С, D, М соответствуют числа 1, 5, 10, 50, 100, 500, 1000. Недостатком этой системы является сложность формальных правил записи чисел и выполнения арифметических действий над ними.

Система счисления называется **позиционной**, если значение каждого знака в числе зависит от позиции, которую занимает знак в записи числа. Это значение находится в однозначной зависимости от позиции, занимаемой цифрой, по некоторому закону. Примером позиционной системы счисления является десятичная система, используемая в повседневной жизни.

Количество различных цифр, употребляемых в позиционной системе, определяет название системы счисления и называется **основанием** системы счисления. Так, в десятичной системе используются десять цифр (от 0 до 9), основанием этой системы является число десять.

В позиционных системах счисления числа записываются в виде последовательности символов:

*N = an an-1 ... a1 a0* , *a-1 a-2 ... а-m (р)* (1)

где *N* – число; *ai* – цифры (символы) числа;

*p* – основание системы счисления;

*n, m* – порядковый номер разряда для целой (*n*) и дробной (*m*) частей числа соответственно.

В этой последовательности запятая отделяет целую часть числа от дробной (коэффициенты при положительных степенях, включая нуль, от коэффициентов при отрицательных степенях). Значение числа, записанного в виде (1), может быть найдено по следующей формуле:

*N = an*·*pn+an-1*·*pn-1+ ... +a0*·*p0+a-1*·*p-1+a-2*·*p-2+ ...+а-m*·*p-m* .(2)

В десятичной системе счисления мы производим вычисления по формуле (2) практически не задумываясь. Возьмём для примера десятичное число 123,45:

2 1 0 -1 -2

123,45 (10) = 1·102+2·101+3·100+4·10-1+5·10-2 = 100+20+3+0,4+0,05

Помимо десятичной, в ЭВМ применяются и другие позиционные системы счисления: двоичная, восьмеричная, шестнадцатеричная. **Двоичная система счисления.**

Используется две цифры: 0 и 1. Особая значимость двоичной системы счисления в информатике определяется тем, что внутреннее представление любой информации в компьютере является двоичным кодом. Примеры представления чисел в двоичной системе счисления представлены в таблице 1. **Восьмеричная система счисления.**

Используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Употреблялась в ЭВМ первого и второго поколений как вспомогательная для записи адресов и данных в сокращенном виде. Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада) (Таблица 1). Триада получается путем добавления, при необходимости, незначащих нулей. **Шестнадцатеричная система счисления.**

Для изображения чисел употребляются 16 цифр. Первые десять цифр этой системы обозначаются цифрами от 0 до 9, а старшие шесть цифр - латинскими буквами: 10-A, 11-B, 12-C, 13-D, 14-E, 15-F. Шестнадцатеричная система используется для записи информации в сокращенном виде. Для представления одной цифры шестнадцатеричной системы счисления используется четыре двоичных разряда (тетрада, или полубайт) (Таблица 1).

**Таблица 1. – Представление чисел в различных системах счисления**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Дес  я  тичная    (  Основ  а  ние 10) | Ри  м  ская | Дв  о  и  ч  ная    (  осн  о  в  а  ние 2) | Вос  ь  мери  ч  ная    (  Основ  а  ние 8) | Дв  о  и  ч  ная    (  три  а  ды) | Шес  т  надцате  -  ри  ч  ная    (  Осн  о  в  а  ние 16) | Двои  ч  ная    (  тетр  а  ды) |
| 0 |  | 0 | 0 | 000 | 0 | 0000 |
| 1 | I | 1 | 1 | 001 | 1 | 0001 |
| 2 | II | 10 | 2 | 010 | 2 | 0010 |
| 3 | III | 11 | 3 | 011 | 3 | 0011 |
| 4 | IV | 100 | 4 | 100 | 4 | 0100 |
| 5 | V | 101 | 5 | 101 | 5 | 0101 |
| 6 | VI | 110 | 6 | 110 | 6 | 0110 |
| 7 | VII | 111 | 7 | 111 | 7 | 0111 |
| 8 | VIII | 1000 | 10 | 001 000 | 8 | 1000 |
| 9 | IX | 1001 | 11 | 001 001 | 9 | 1001 |
| 10 | X | 1010 | 12 | 001 010 | A | 1010 |
| 11 | XI | 1011 | 13 | 001 011 | B | 1011 |
| 12 | XII | 1100 | 14 | 001 100 | C | 1100 |
| 13 | XIII | 1101 | 15 | 001 101 | D | 1101 |
| 14 | XIV | 1110 | 16 | 001 110 | E | 1110 |
| 15 | XV | 1111 | 17 | 001 111 | F | 1111 |
| 16 | XVI | 10000 | 20 | 010 000 | 10 | 0001 0000 |
| 17 | XVII | 10001 | 21 | 010 001 | 11 | 0001 0001 |

**1.2 ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ**

**Перевод чисел в десятичную систему** осуществляется путем составления степенного ряда (2) с основанием той системы, из которой число переводится. Затем подсчитывается значение суммы. ***Пример.***

а) Перевести 10101101,101(2) в десятичную систему счисления

10101101,101(2) = 1·27 + 0·26 + 1·25 + 0·24 + 1·23 + 1·22 + 0·21 + 1·20 + 1·2-1

+ + 0·2-2 + 1·2-3 = 173,625(10)

б) Перевести 703,04(8) в десятичную систему счисления 703,04(8) = 7·82 + 0·81 + 3·80+ 0·8-1 + 4·8-2 = 451,0625(10)

в) Перевести B2E,4(16) в десятичную систему счисления B2E,4(16) = 11·162 + 2·161 + 14·160 + 4·16-1 = 2862,25(10)

**Перевод целых десятичных чисел в недесятичную систему** **счисления** осуществляется последовательным делением десятичного числа на основание той системы, в которую оно переводится, до тех пор, пока не получится частное, меньшее этого основания. Число в новой системе записывается в виде остатков деления, начиная с последнего.

***Пример.***

а) Перевести 181(10) в восьмеричную систему счисления

\_181 8

176 \_22 8

**5** 16 **2**

**6**

Результат: 181(10) = 265(8)

б) Перевести 622(10) в шестнадцатеричную систему счисления

\_622 16

48 \_38 16

\_142 32 **2**

128 **6**

**14**

Результат: 622(10) = 26E(16)

**Перевод правильных дробей из десятичной системы счисления в недесятичную.** Для перевода правильной десятичной дроби в другую систему эту дробь надо последовательно умножать на основание той системы, в которую она переводится. При этом умножаются только дробные части. Дробь в новой системе записывается в виде целых частей произведений, начиная с первого.

***Пример.***

Перевести 0,3125(10) в восьмеричную систему счисления

0 , 3125 × 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **2** | 5000 × 8 | | **4** | 0000 | |

Результат: 0,3125(10) = 0,24(8)

***Замечание.*** Конечной десятичной дроби может соответствовать бесконечная (периодическая) дробь в недесятичной системе счисления. В этом случае количество знаков в представлении дроби в новой системе берется в зависимости от требуемой точности.

***Пример.***

Перевести 0,65(10) в двоичную систему счисления с точностью до 6 зна-

ков после запятой.

0, 65 × 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **1** | 3 × 2 | | **0** | 6 × 2 | | **1** | 2 × 2 | | **0** | 4 × 2 | | **0** | 8 × 2 | | **1** | 6 × 2 | |

...

Результат: 0,65(10) ≈ 0,101001 (2)

**Для перевода неправильной десятичной дроби в систему счисления с недесятичным основанием** необходимо отдельно перевести целую часть и отдельно дробную.

***Пример.***

Перевести 23,125(10) в двоичную систему счисления

1) Переведем целую часть:

\_ 23 2 22 \_11 2

**1** 10 \_5 2

**1** 4 \_2 2

**1** 2 **1**

# 0

2) Переведем дробную часть:

0, 125 × 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **0** | 25 × 2 | | **0** | 5 × 2 | | **1** | 0 | |

Таким образом: 0,125(10) = 0,001(2); 23(10) = 101112. Результат: 23,125(10) = 10111,001(2).

Необходимо отметить, что целые числа остаются целыми, а правильные дроби - дробями в любой системе счисления.

**Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму** достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) – для восьмеричной системы счисления или четырехразрядным двоичным числом (тетрадой) – для шестнадцатеричной системы счисления (таблица 1), после чего отбрасывают незначащие нули в старших и младших разрядах.

***Пример.***

а) Перевести 305,4(8) в двоичную систему счисления

3 0 5 , 4 (8)

011 000 101 100 = 11000101,1(2)

б) Перевести 7B2,E(16) в двоичную систему счисления

7 В 2 , Е (16)

0111 1011 0010 1110 = 11110110010,111(2)

**Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе** поступают следующим образом: двигаясь от десятичной точки влево и вправо, разбивают двоичное число на группы по три (четыре) разряда, дополняя при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой (таблица 1).

***Пример.***

а) Перевести 1101111001,1101(2) в восьмеричную систему счисления

001 101 111 001 , 110 100 = 1571,64(8)

1 5 7 1 6 4

б) Перевести 11111111011,100111(2) в шестнадцатеричную систему счисления

0111 1111 1011 , 1001 1100 = 7FB,9C(16)

7 F B 9 С

**Перевод из восьмеричной в шестнадцатеричную систему и обратно** удобно осуществлять через двоичную систему с помощью триад и тетрад.

***Пример.***

Перевести 175,24(8) в шестнадцатеричную систему счисления

1 7 5 , 2 4 (8) =1111101,0101(2)=0111 1101 , 0101 (2) = 7D,5(16)

001 111 101 010 100 7 D 5

##### 1.3 ДВОИЧНАЯ АРИФМЕТИКА

Правила выполнения арифметических действий над двоичными числами такие же, как и в десятичной системе, и задаются таблицами двоичного сложения, вычитания и умножения (таблица 2). Подобные таблицы для восьмеричной и шестнадцатеричной систем счисления приведены в Приложении.

**Таблица 2 - Арифметические действия над двоичными числами**

|  |  |  |
| --- | --- | --- |
| **Таблица двоичного сложения** | **Таблица двоичного вычитания** | **Таблица двоичного умножения** |
| 0+0=0 0+1=1  1+0=1  1+1=10 | 0-0=0 1-0=1  1-1=0  10-1=1 | 0×0=0 0×1=0  1×0=0 1×1=1 |

При сложении двоичных чисел производится сложение цифр слагаемых в каждом разряде и единиц переноса из соседнего младшего разряда, если они имеются. При этом необходимо учитывать, что в двоичной системе переполнение разряда наступает при количестве единиц, больше либо равным двум. В случае переполнения нужно вычесть из текущего разряда число, равное основанию системы (в данном случае – два), и добавить единицу переноса в следующий старший разряд.

Прежде чем рассматривать приведенные ниже примеры, полезно попробовать получить для различных систем счисления порядковые последовательности путём прибавления единицы к предыдущему числу, начиная с нуля, а затем сравнить их с соответствующими столбцами таблицы 1. Затем попробуйте получить последовательности путём вычитания в обратном порядке.

***Пример.***

Выполнить сложение двоичных чисел (X, Y, Z):

а) X=1101, Y=101;

единицы переноса

1 1 1 1 0 1

+

1 0 1

1 0 0 1 0

Результат: 1101+101=10010.

б) X=1101, Y=101, Z=111;

1

1 1 1

1 1 0 1

+

1 0 1

+

1 1 1

1 1 0 0 1

Результат: 1101+101+111=11001.

При вычитании двоичных чисел, аналогично вычитанию десятичных, может возникнуть необходимость займа единицы из предыдущего старшего разряда. Эта занимаемая единица переносится в текущий разряд как двойка (количество единиц, равное основанию).

***Пример.***

Заданы двоичные числа X=10010 и Y=101. Вычислить X - Y.

. . .

\_1 0 0 1 0

### 1 0 1

1. 1 1 0 1

Результат 10010 - 101=1101.

Умножение двоичных чисел оказывается гораздо проще десятичных и

сводится к операциям сдвига и сложения.

***Пример.*** Заданы двоичные числа X=1001 и Y=101. Вычислить X × Y.

1. 0 0 1

# 1 0 1

1 0 0 1

1 0 0 1

1 0 1 1 0 1

Результат: 1001×101=101101.

Выполнение деления в двоичной системе также проще, чем в десятич-

ной, и сводится к операциям сравнения, сдвига и вычитания.

***Пример.***

Заданы двоичные числа X=1100,011 и Y=10,01. Вычислить X : Y

|  |  |
| --- | --- |
| 1 0 0 | 1 |
| 1 0 1, | 1 |

\_ 1 1 0 0 0 1, 1

1 0 0 1

# \_ 1 1 0 1

1 0 0 1 \_ 1 0 0 1 1 0 0 1

### 0

Результат. 1100,011 : 10,01=101,1.

##### 1.4 ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ

В цифровых ЭВМ числовая информация представляется в двух формах:

* с фиксированной точкой (естественная форма);
* с плавающей точкой (экспоненциальная форма).

При представлении чисел с фиксированной точкой подразумевается, что положение точки, разделяющей число на целую и дробную части, неизменно для всех чисел. Эта форма наиболее проста, естественна, но имеет небольшой диапазон представления чисел и поэтому не всегда приемлема при вычислениях. В современных ЭВМ естественная форма используется, например, для представления целых чисел (дробная часть числа всегда отсутствует), денежных сумм (дробная часть всегда составляет четыре знака).

Представление с плавающей точкой любого числа *N* в общем виде опи-

сывается следующей формулой:

*N =* ± *M* ⋅ *p*±*k*,(3)

где ±*M* – мантисса (дробная часть) числа; *p* – основание системы счисления;

±*k* – порядок (целое число), при этом положительный знак мантиссы и порядка может опускаться, а

при указании порядка в десятичной системе принято использовать символ Е. Например, десятичное число с фиксированной точкой 123,45 может быть представлено в форме с плавающей точкой как 0,12345 ⋅ 103, или, как это принято, 1.2345Е+02. Такая форма представления имеет огромный диапазон отображения чисел и является основной в современных ЭВМ.

Исходные данные в ЭВМ хранятся в виде двоичных чисел, то есть записываются в виде последовательности нулей и единиц. В памяти ЭВМ одна двоичная цифра записывается в один двоичный разряд, называемый **битом**. За единицу представления данных в цифровых ЭВМ принят **байт** – восемь бит, поэтому число разрядов ячеек памяти всегда кратно восьми, а данные имеют байтовую структуру, то есть состоят из определенного числа байтов.

Более крупные единицы измерения данных образуются добавлением

префиксов *кило-, мега-, гига-, тера-*:

1 килобайт (Кбайт) = 210 байт = 1024 байт

1 мегабайт (Мбайт) = 220 байт = 1024 Кбайт

1 гигабайт (Гбайт) = 230 байт = 1024 Мбайт

1 терабайт (Тбайт) = 240 байт = 1024 Гбайт

Для представления положительных и отрицательных чисел в машинах используются специальные коды: *прямой, обратный* и *дополнительный.* Причем два последних позволяют заменить неудобную для ЭВМ операцию вычитания на операцию сложения с отрицательным числом; дополнительный код обеспечивает более быстрое выполнение операций при помощи сумматора, поэтому в ЭВМ применяется чаще именно он. Рассмотрим правила кодирования на примере целых чисел.

Для перевода числа в **прямой код** знак числа опускается, а в старший (знаковый) разряд ставится 0, если число положительное, и 1 – если число отрицательное. Младшие разряды кода являются двоичным представлением модуля числа. Оставшиеся разряды кода заполняются нулями. Отметим, что перевод положительных чисел в прямой, обратный и дополнительный код не изменяет изображения этих чисел (таблица 3).

Для перевода отрицательного числа в **обратный код** необходимо все, кроме знакового, разряды прямого кода проинвертировать (заменить нули на единицы, а единицы на нули).

Для перевода отрицательного числа в **дополнительный код** необходимо к младшему разряду его обратного кода прибавить единицу.

Перевод отрицательного числа из дополнительного кода в прямой осуществляется в обратной последовательности: сначала вычитается единица, затем инвертируются разряды. Напоминаем, что положительное число (0 в старшем разряде) обратному переводу не подвергается, поскольку имеет одинаковую запись как в прямом коде, так и в дополнительном.

**Таблица 3 – Примеры представления целых чисел в шестнадцатиразрядных двоичных кодах**

|  |  |  |  |
| --- | --- | --- | --- |
| **Число** | **прямой код** | **обратный код** | **дополнительный код** |
| 0 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 |
| 1 | 0000 0000 0000 0001 | 0000 0000 0000 0001 | 0000 0000 0000 0001 |
| -1 | 1000 0000 0000 0001 | 1111 1111 1111 1110 | 1111 1111 1111 1111 |
| 20 | 0000 0000 0001 0100 | 0000 0000 0001 0100 | 0000 0000 0001 0100 |
| -20 | 1000 0000 0001 0100 | 1111 1111 1110 1011 | 1111 1111 1110 1100 |

При написании программ важно определить диапазоны значений и формы представления обрабатываемой информации. Например, в языках программирования семейства BASIC (Бейсик) типы переменных INTEGER и LONG используются, соответственно, для хранения целых чисел со знаком в шестнадцатиразрядном (два байта, или полуслово) и тридцатидвухразрядном (четыре байта, или машинное слово) дополнительном коде. Знак числа фиксируется в нулевом бите первого байта (крайний левый бит). Цифровая часть числа хранится в остальных битах поля числа, причем младший двоичный разряд числа находится в последнем, правом бите последнего байта. Переменные типа SINGLE и DOUBLE используются для хранения чисел с плавающей точкой в четырех или восьми байтах (двойное слово) соответственно. Знак хранится в нулевом бите, биты 1-7 отводятся под порядок числа, остальные биты используются для разрядов мантиссы. Как правило, мантисса хранится в нормальном виде, то есть первым ее разрядом не является 0.

##### 1.5 КОДИРОВАНИЕ ИНФОРМАЦИИ В ЭВМ

Для представления символьной информации в двоичной форме используются таблицы кодировки. При длине кода один байт (8 бит) можно закодировать 256 (то есть 28) различных символов. Этого достаточно для кодирования символов любого национального алфавита, но недостаточно, чтобы представить в одной таблице символы всех алфавитов. Уровня международного стандарта достигла система **ASCII** (American Standard Code for Information Interchange —

Американский стандартный код для обмена информацией). Эта система устанавливает две таблицы кодирования: *базовую* и *расширенную*. В базовой таблице закреплены значения кодов от 0 до 127. Первые 32 кода являются *управляющими*, они предназначены для управления устройствами вывода данных и определяются производителями. Большинство значений кодов базовой таблицы ASCII представлено в Приложении.

Расширенная таблица определяет значения кодов со 128 по 255 и используется национальными системами кодирования. Например, в России наибольшее распространение получили три разных системы: *ГОСТ-альтернативная* (на компьютерах, работающих в операционной системе MS-DOS); *Windows-1251*; *КОИ-8* (код обмена информации, восьмизначный).

В настоящее время всё большее распространение получает *универсальная* система кодирования *(UNICODE)*. В ней используется шестнадцатиразрядный код, позволяющий представить 65 536 (то есть 216) символов. Этого достаточно для кодирования символов большинства языков планеты. Однако текст в кодировке UNICODE занимает вдвое больший объем в памяти ЭВМ или на машинном носителе информации, по сравнению с этим же текстом в кодировке ASCII или любой другой восьмиразрядной системе кодирования.

Существует множество способов представления графики, звука, видео, других видов данных и их совокупностей, и оно постоянно расширяется. Таким образом, многообразие систем кодирования ставит одну из важнейших задач информатики – задачу межсистемного преобразования данных. **1.6 ЗАДАНИЯ К РАБОТЕ**

Ц е л ь р а б о т ы . Изучение систем счисления, используемых в вычислительной технике и правил перевода чисел из одной системы счисления в другую.

З а д а н и е к р а б о т е с о с т о и т и з ч е т ы р ё х п у н к т о в .

1. Перевести данные числа в десятичную систему счисления.
2. Перевести данные числа из десятичной системы счисления в двоич-ную, восьмеричную и шестнадцатеричную системы счисления (с точностью 6 знаков после запятой). Выполнить проверку путем обратного перевода в десятичную систему счисления.
3. Сложить числа в указанной системе счисления.
4. Выполнить вычитание в указанной системе счисления.

### **Вариант 1**

1. а) 10100010(2); б) 1110010111(2); в) 110010010,101(2);

г) 1111011100,10011(2); д) 605,02(8); е) 3C8,8(16).

1. а) 969(10); б) 549(10); в) 973,375(10); г) 508,5(10); д) 281,09(10).
2. а) 1111010100(2)+10000000010(2); б) 101001011(2)+10000000010(2);

в) 1011101001,1(2)+1110111,01(2); г) 1053,34(8)+1513,2(8); д) 40A,E8(16)+92,7(16).

1. а) 1001100011(2)-111111110(2); б) 1110001000(2)-1011110(2);

в) 10000010111,001(2)-1000010,01(2); г) 553,2(8)-105,5(8); д) 298,9(16)-67,4(16). **Вариант 2**

1. а) 1100111011(2); б) 10000000111(2); в) 10110101,1(2);

г) 100000110,10101(2); д) 671,24(8); е) 41A,6(16).

1. а) 666(10); б) 305(10); в) 153,25(10); г) 162,25(10); д) 248,46(10)
2. а) 10000011(2)+1000011(2); б) 1010010000(2)+1101111011(2);

в) 110010,101(2)+1011010011,01(2); г) 356,5(8)+1757,04(8); д) 293,8(16)+3CC,98(16).

1. а) 100111001(2)-110110(2); б) 1111001110(2)-111011010(2);

в) 1101111011,01(2)-101000010,0111(2); г) 2025,2(8)-131,2(8); д) 2D8,4(16)-A3,B(16). **Вариант 3**

1. а) 1100000000(2); б) 1101011111(2); в) 1011001101,00011(2);

г) 1011110100,011(2); д) 1017,2(8); е) 111,B(16).

1. а) 273(10); б) 661(10); в) 156,25(10); г) 797,5(10); д) 53,74(10)
2. а) 1110001000(2)+110100100(2); б) 1001001101(2)+1111000(2);

в) 111100010,0101(2)+1111111,01(2); г) 573,04(8)+1577,2(8); д) 108,8(16)+21B,9(16).

1. а) 1010111001(2)-1010001011(2); б) 1110101011(2)-100111000(2);

в) 1110111000,011(2)-111001101,001(2); г) 1300,3(8)-464,2(8); д) 37C,4(16)-1D0,2(16). **Вариант 4**

1. а) 1100001001(2); б) 1100100101(2); в) 1111110110,01(2);

г) 11001100,011(2); д) 112,04(8); е) 334,A(16).

1. а) 105(10); б) 358(10); в) 377,5(10); г) 247,25(10); д) 87,27(10)
2. а) 101000011(2)+110101010(2); б) 111010010(2)+1011011110(2);

в) 10011011,011(2)+1111100001,0011(2); г) 1364,44(8)+1040,2(8); д) 158,A(16)+34,C(16).

1. а) 1111111000(2)-100010011(2); б) 1111101110(2)-11100110(2);

в) 1001100100,01(2)-10101001,1(2); г) 1405,3(8)-346,5(8); д) 3DD,4(16)-303,A(16). **Вариант 5**

1. а) 1101010001(2); б) 100011100(2); в) 1101110001,011011(2);

г) 110011000,111001(2); д) 1347,17(8); е) 155,6C(16).

1. а) 500(10); б) 675(10); в) 810,25(10); г) 1017,25(10); д) 123,72(10)
2. а) 1000101101(2)+1100000010(2); б) 1111011010(2)+111001100(2);

в) 1001000011,1(2)+10001101,101(2); г) 415,24(8)+1345,04(8); д) 113,B(16)+65,8(16).

1. а) 1101111100(2)-100100010(2); б) 1011010110(2)-1011001110(2);

в) 1111011110,1101(2)-1001110111,1(2); г) 1333,2(8)-643,2(8); д) 176,7(16)-E5,4(16). **Вариант 6**

1. а) 111000100(2); б) 1011001101(2); в) 10110011,01(2);

г) 1010111111,011(2); д) 1665,3(8); е) FA,7(16).

1. а) 218(10); б) 808(10); в) 176,25(10); г) 284,25(10); д) 253,04(10)
2. а) 11100000(2)+1100000000(2); б) 110101101(2)+111111110(2);

в) 10011011,011(2)+1110110100,01(2); г) 1041,2(8)+1141,1(8); д) 3C6,8(16)+B7,5(16).

1. а) 10110010(2)-1010001(2); б) 1101000000(2)-10000000(2);

в) 1100101111,1101(2)-100111000,1(2); г) 1621,44(8)-1064,5(8); д) 1AC,B(16)-BD,7(16). **Вариант 7**

1. а) 1111000111(2); б) 11010101(2); в) 1001111010,010001(2);

г) 1000001111,01(2); д) 465,3(8); е) 252,38(16).

1. а) 306(10); б) 467(10); в) 218,5(10); г) 667,25(10); д) 318,87(10)
2. а) 1000001101(2)+1100101000(2); б) 1010011110(2)+10001000(2);

в) 1100111,00101(2)+101010110,011(2); г) 520,4(8)+635,4(8); д) 2DB,6(16)+15E,6(16).

1. а) 1101000101(2)-111111000(2); б) 11110101(2)-110100(2);

в) 1011101011,001(2)-1011001000,01001(2); г) 1034,4(8)-457,44(8); д) 239,A(16)-9C,4(16). **Вариант 8**

1. а) 110010001(2); б) 100100000(2); в) 1110011100,111(2);

г) 1010111010,1110111(2); д) 704,6(8); е) 367,38(16).

1. а) 167(10); б) 113(10); в) 607,5(10); г) 828,25(10); д) 314,71(10)
2. а) 10101100(2)+111110010(2); б) 1000000010(2)+110100101(2); в) 1110111010,10011(2)+1011010011,001(2); г) 355,2(8)+562,04(8); д) 1E5,18(16)+3BA,78(16).
3. а) 1010110010(2)-1000000000(2); б) 1111100110(2)-10101111(2);

в) 1101001010,101(2)-1100111000,011(2); г) 1134,54(8)-231,2(8); д) 2DE,6(16)-12A,4(16). **Вариант 9**

1. а) 1000110110(2); б) 111100001(2); в) 1110010100,1011001(2);

г) 1000000110,00101(2); д) 666,16(8); е) 1C7,68(16).

1. а) 342(10); б) 374(10); в) 164,25(10); г) 520,375(10); д) 97,14(10).
2. а) 1101010000(2)+1011101001(2); б) 100000101(2)+1100001010(2);

в) 1100100001,01001(2)+1110111111,011(2); г) 242,2(8)+1153,5(8); д) 84,8(16)+27E,8(16).

1. а) 1111110(2)-1111011(2); б) 1111100000(2)-111110011(2);

в) 1111011111,1001(2)-1010111100,01(2); г) 1241,34(8)-1124,3(8); д) 15F,A(16)-159,4(16). **Вариант 10**

1. а) 101111111(2); б) 1111100110(2); в) 10011000,1101011(2);

г) 1110001101,1001(2); д) 140,22(8); е) 1DE,54(16).

1. а) 524(10); б) 222(10); в) 579,5(10); г) 847,625(10); д) 53,35(10).
2. а) 1101010000(2)+11100100(2); б) 100110111(2)+101001000(2);

в) 1111100100,11(2)+1111101000,01(2); г) 1476,3(8)+1011,1(8); д) 3E0,A(16)+135,8(16).

1. а) 1010010100(2)-11101110(2); б) 10000001110(2)-10011100(2);

в) 1110100111,01(2)-110000001,1(2); г) 1542,5(8)-353,24(8); д) 3EB,8(16)-3BA,8(16). **Вариант 11**

1. а) 11101000(2); б) 1010001111(2); в) 1101101000,01(2);

г) 1000000101,01011(2); д) 1600,14(8); е) 1E9,4(16).

1. а) 113(10); б) 875(10); в) 535,1875(10); г) 649,25(10); д) 6,52(10).
2. а) 1000111110(2)+1011000101(2); б) 1001000(2)+1101101001(2);

в) 110110010,011(2)+1000011111,0001(2); г) 620,2(8)+1453,3(8); д) 348,1(16)+234,4(16).

1. а) 1100001010(2)-10000011(2); б) 1101000001(2)-10000010(2);

в) 110010110,011(2)-10010101,1101(2); г) 1520,5(8)-400,2(8); д) 368,4(16)-239,6(16).

**Вариант 12** 1. а) 10000011001(2); б) 10101100(2); в) 1101100,01(2); г) 1110001100,1(2); д) 1053,2(8); е) 200,6(16).

1. а) 294(10); б) 723(10); в) 950,25(10); г) 976,625(10); д) 282,73(10).
2. а) 1000111110(2) + 10111111(2); б) 1111001(2) + 110100110(2); в) 1001110101,00011(2) + 1001001000,01(2); г) 104,4(8) + 1310,62(8); д) 2BD,3(16)+EB,C(16).
3. а) 11110111(2)-11110100(2); б) 1001100111(2)-101100111(2);

в) 1100110111,001(2)-1010001101,0011(2); г) 631,1(8)-263,2(8); д) 262,8(16)-1D6,88(16). **Вариант 13**

1. а) 110111101(2); б) 1110011101(2); в) 111001000,01(2);

г) 1100111001,1001(2); д) 1471,17(8); е) 3EC,5(16).

1. а) 617(10); б) 597(10); в) 412,25(10); г) 545,25(10); д) 84,82(10).
2. а) 1110100100(2)+1010100111(2); б) 1100001100(2)+1010000001(2); в) 1100111101,10101(2) + 1100011100,0011(2); г) 750,16(8) + 1345,34(8); д) 158,4(16)+396,8(16).
3. а) 10000000010(2)-100000001(2); б) 1110111111(2)-1010001(2);

в) 1011001100,1(2)-100100011,01(2); г) 1110,62(8)-210,46(8); д) 1D8,D8(16)-110,4(16). **Вариант 14**

1. а) 1101100000(2); б) 100001010(2); в) 1011010101,1(2);

г) 1010011111,1101(2); д) 452,63(8); е) 1E7,08(16).

1. а) 1047(10); б) 335(10); в) 814,5(10); г) 518,625(10); д) 198,91(10).
2. а) 1101100101(2)+100010001(2); б) 1100011(2)+110111011(2);

в) 1010101001,01(2)+10011110,11(2); г) 1672,2(8)+266,2(8); д) 18B,A(16)+2E9,2(16).

1. а) 1110111011(2)-100110111(2); б) 1110000101(2)-1001110(2);

в) 1011110100,0011(2)-101001011,001(2); г) 1560,22(8)-1142,2(8); д) 1A5,8(16)-7D,A(16). **Вариант 15**

1. а) 1001110011(2); б) 1001000(2); в) 1111100111,01(2);

г) 1010001100,101101(2); д) 413,41(8); е) 118,8C(16).

1. а) 164(10); б) 255(10); в) 712,25(10); г) 670,25(10); д) 11,89(10)
2. а) 1100001100(2)+1100011001(2); б) 110010001(2)+1001101(2);

в) 111111111,001(2)+1111111110,0101(2); г) 1443,1(8)+242,44(8); д) 2B4,C(16)+EA,4(16).

1. а) 1001101100(2)-1000010111(2); б) 1010001000(2)-1000110001(2);

в) 1101100110,01(2)-111000010,1011(2); г) 1567,3(8)-1125,5(8); д) 416,3(16)-255,3(16). **Вариант 16**

1. а) 1010100001(2); б) 10000010101(2); в) 1011110000,100101(2);

г) 1000110001,1011(2); д) 1034,34(8); е) 72,6(16).

1. а) 887(10); б) 233(10); в) 801,5(10); г) 936,3125(10); д) 218,73(10).
2. а) 1010110101(2)+101111001(2); б) 1111100100(2)+100110111(2);

в) 111111101,01(2)+1100111100,01(2); г) 106,14(8)+322,5(8); д) 156,98(16)+D3,2(16).

1. а) 1111100100(2)-110101000(2); б) 1110110100(2)-1101010101(2);

в) 1100001,0101(2)-1011010,101(2); г) 537,24(8)-510,3(8); д) 392,B(16)-149,5(16).

**1.7 ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ** З а д а н и е

1. Перевести данные числа в десятичную систему счисления:

а) 1110010011,1011(2); б) 772,24(8); в) 81,A(16).

1. Перевести число 119(10) из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления. Выполнить проверку путем обратного перевода в десятичную систему счисления.
2. Сложить числа:

а) 1101011110,001(2) + 111100001,011(2); б) 1034,16(8) + 205,2(8); в) 33C,2(16)+37D,4(16). 4. Выполнить вычитание а) 110001100,011(2) - 1101100,11(2); б) 1733,3(8) - 355,2(8); в) 26F,4(16)-D3,6(16).

Р а с ч е т з а д а н и я

1. а)1110010011,1011(2) = 1·29 + 1·28 + 1·27 + 0·26 + 0·25 + 1·24 + 0·23 +

+ 0·22  +1·21 + 1·20 + 1·2-1 + 0·2-2 + 1·2-3 + 1·2-4 = 512 + 256 + 128 +

+ 0 + 0 + 16 + 0 + 0 + 2 + 1 + 0,5 + 0 + 0,125 +0,0625 = 915,6875(10)

б) 772,24(8) = 7·82 + 7·81 + 2·80 + 2·8-1 + 4·8-2 = 448 + 56 + 2 + 0,25 +

+ 0,0625= 506,3125(10)

в) 81,А(16) = 8·161 + 1·160 + 10·16-1 = 128 + 1 + 0,625 = 129,625(10)

2. 119

(10)

\_

119

2

118

\_59

2

**1**

58

\_29

2

**1**

28

\_14

2

**1**

14

\_7

2

**0**

6

\_3

2

**1**

2

**1**

**1**

Проверка:

1110111(2)=1·20+1·21+1·22+0·23+1·24+1·25+1·26=1+2+4+16+32+64=119(10)

**\_** 119 8

112 \_14 8

**7** 8 **1**

**6**

Проверка:167(8) = 7·80+6·81+1·82 = 7+48+64 = 119

|  |  |
| --- | --- |
| 119 112 | 16 |
| **7** |

\_

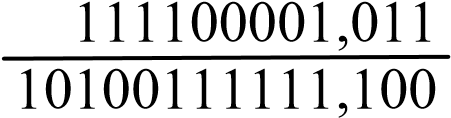
**7**

Проверка:77(16) = 7·160+7·161 = 7+112 =119

119(10) = 1110111(2) = 167(8) = 77(16)

1. а) 1101011110,001(2) + 111100001,011(2)  = 10100111111,1(2)

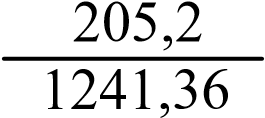
+1101011110,001



б) 1034,16(8) + 205,2(8) = 1241,36(8)

.

+1034,16



в) 33С,2(16) + 37D,4(16) = 6В9,6(16)

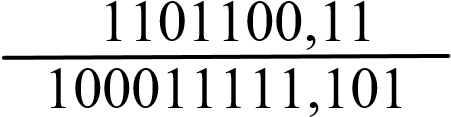
.

+33С,2 37D,4 6В9,6

1. а) 110001100,011(2) – 1101100,11(2) = 100011111,101(2)

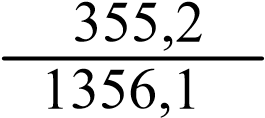
. . . . . . . .

* 110001100,011



б) 1733,3(8) – 355,2(8) = 1356,1(8)

* 1733,3



в) 26F,4(16) – D3,6(16) = 19В,Е(16)

* 26F,4 D3,6

19В,Е

#### 2 ЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ЭВМ

##### 2.1 ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

Слово «логика» означает как совокупность правил, которым подчиняется процесс мышления, так и науку о правилах рассуждений. Логика, как наука о законах и формах мышления, изучает абстрактное мышление как средство познания объективного мира.

Основными формами абстрактного мышления являются:

— ПОНЯТИЯ,

— СУЖДЕНИЯ, — УМОЗАКЛЮЧЕНИЯ.

ПОНЯТИЕ — форма мышления, в которой отражаются существенные признаки отдельного предмета или класса однородных предметов, например: «портфель»; «трапеция»; «ветер».

СУЖДЕНИЕ — мысль, в которой что-либо утверждается или отрицается о предметах. Суждения являются истинными или ложными повествовательными предложениями. Они могут быть простыми и сложными. Например: «Весна наступила»; «Грачи прилетели»; «Весна наступила, и грачи прилетели».

УМОЗАКЛЮЧЕНИЕ — прием мышления, посредством которого из исходного знания получается новое знание; из одного или нескольких истинных суждений, называемых посылками, мы по определенным правилам вывода получаем заключение.

Все металлы — простые вещества.

Литий — металл.

Литий — простое вещество.

Чтобы достичь истины при помощи умозаключений, надо соблюдать законы логики. Существует формальная и математическая логика.

**Формальная логика** — наука о законах и формах мышления.

**Математическая логика** изучает логические связи и отношения, лежащие в основе дедуктивного (логического) вывода.

Формальная логика связана с анализом наших обычных содержательных умозаключений, выражаемых разговорным языком. Математическая логика изучает только умозаключения со строго определенными объектами и суждениями, для которых можно однозначно решить, истинны они или ложны.

В основе логических схем и устройств ЭВМ лежит специальный аппарат, использующий законы математической логики. Математическая логика изучает вопросы применения математических методов для решения логических задач и построения логических схем. Знание логики необходимо при разработке алгоритмов и программ, так как в большинстве языков программирования есть логические операции.

**Алгебра логики** — это раздел математической логики, значения всех элементов (функций и аргументов) которой определены в двухэлементном множестве: «Истина» («True») и «Ложь» («False»), или 1 и 0.

В математической логике суждения называются высказываниями. Алгебру логики иначе называют алгеброй высказываний.

**Высказывание** — это повествовательное предложение, о котором можно сказать, истинно оно или ложно.

Примеры высказываний:

Сейчас идет снег. *может быть истинным или ложным*

Земля — планета Солнечной системы. *истинно*

2 + 8 < 5 *ложно*

5 \* 5 = 25 *истинно*

Всякий квадрат есть параллелограмм. *истинно*

Всякий параллелограмм есть квадрат. *ложно*

2 \* 2 = 5 *ложно*

А вот примеры, не являющиеся высказываниями: «Уходя, гасите свет!»;

«Да здравствует мыло душистое и полотенце пушистое!»

Высказывания, приведенные выше, являются простыми. Сложные высказывания получаются путем объединения простых высказываний связками — союзами И, ИЛИ и частицей НЕ. Значение истинности сложных высказываний зависит от истинности входящих в них простых высказываний и от объединяющих их связок.

##### 2.2 ОПЕРАЦИИ СРАВНЕНИЯ

Операции сравнения называют еще *операциями отношения* (relation operations), поскольку в них осуществляется оценка взаимосвязи (отношений) двух операндов. Под *операндом* понимается то, над чем выполняется операция.

В таблице 6 перечислены *операторы*, используемые для обозначения операций сравнения в языках программирования. Результат сравнения может быть либо истинным, либо ложным (Тruе или False). Приоритет операций сравнения ниже, чем у арифметических операций.

**Таблица 6 – Примеры операторов сравнения в разных языках программирования**

|  |  |  |
| --- | --- | --- |
| **Операция сравнения** | **Оператор** | |
| **Basic, Pascal** | **С++, Java** |
| равно | **=** | **==** |
| не равно | **<>** | **!=** |
| больше | **>** | **>** |
| меньше | **<** | **<** |
| больше или равно | **>=** | **>=** |
| меньше или равно | **<=** | **<=** |

##### 2.3 ЛОГИЧЕСКИЕ ОПЕРАЦИИ

Логические операции иногда называют булевыми, так как взяты они из булевой алгебры. Операндами в них выступают булевы величины и возвращают они тоже булевы значения. В таблице 7 приведены основные логические операторы, используемые в разных языках программирования, в порядке убывания приоритета. При этом следует отметить, что в разных языках программирования логические операции могут иметь разный приоритет по отношению к другим операциям – арифметическим и сравнения. Например, в Бейсике приоритет логических операций ниже, чем операций сравнения. В Паскале – наоборот, а приоритет оператора not даже выше, чем у арифметических операций.

**Таблица 7 –Приоритет основных логических операторов**

|  |  |  |
| --- | --- | --- |
| **Название логической операции** | **Оператор** | |
| **Basic, Pascal** | **С++, Java** |
| логическое отрицание «НЕ» (инверсия) | Not | ! |
| логическое «И» (конъюнкция) | And | && |
| логическое «ИЛИ» (дизъюнкция) | Or | | | |

Простейший логический оператор — NOT. Он предшествует единственному операнду и возвращает его логическую противоположность — иначе говоря, «отрицает» операнд. Например, выражение *NOT False* дает в результате Тruе (булево значение, обратное False); а выражение NOT *(4 < 3) —* Тruе, так как условие *(4 < 3)* ложно. Кроме того, этот оператор обладает наивысшим приоритетом по сравнению с остальными булевыми операторами. Кстати, операция отрицания является *унарной*, или *одноместной*, в отличие от остальных логических операций, являющихся *бинарными*, или *двуместными*, то есть выполняющими действия с двумя операндами.

Следующий (в порядке убывания приоритета) оператор — AND. Он возвращает True, только если оба операнда имеют значение True. По смыслу он соответствует английскому слову *and* (русскому *и),* что можно проиллюстрировать, например, таким утверждением: «Он получит эту должность, только если знает английский *и* разбирается в компьютерах».

Оператор OR возвращает True, если любой из двух операндов True или если оба True. И в этом случае логический смысл оператора совпадает с соответствующим словом в языке: «Я куплю этого кота, если он не дороже 50 долларов ***или*** если он — сиамской породы».

Аналогично операциям сравнения логические операции могут связать два или более отношения и возвратить истинную или ложную величину, используемую для управления ходом выполнения программы.

***Пример 1.*** Определить, что сумма баллов S**,** набранная студентом на

тестированиинаходится в пределах 60÷100 баллов, то есть принадлежит интервалу [60,100] 60 100 :

S >= 60 AND S <= 100.

***Пример 2.*** Определить, что сумма баллов S**,** набранная студентом на

тестированиине входит в пределы 60÷100 баллов, то есть находится вне интервала [60,100]60 100 :

NOT (S >= 60 AND S <= 100); или, что даёт тот же самый результат, S < 60 OR S > 100.

Правила выполнения операций в алгебре логики определяются рядом аксиом, теорем и следствий. В частности, для алгебры логики выполняются следующие законы:

1. сочетательный:

*(а OR b) OR с = а OR (b OR с); (а AND b) AND с = а AND (b AND с);*

1. переместительный:

*а OR b = b OR а; а AND b = b AND а;*

1. распределительный:

*а AND (b OR с) = а AND b OR а AND с; а OR b AND с = (а OR b) AND (а OR с).*

Справедливы также следующие соотношения: *NOT NOT a = a; а OR а AND b = а; NOT a OR NOT b = NOT (a AND b)* и другие.

Логические операции могут производиться не только над булевыми величинами, но и над битами операндов. В этом случае логическая операция возвращает поразрядный результат, который либо истинен (1), либо ложен (0). В языках программирования могут существовать специальные операторы побитового выполнения логических операций. Например, в «Си++» и «Ява» поразрядным (побитовым) операциям НЕ, И, ИЛИ соответствуют операторы ~, &, | (сравните с операторами таблицы 7).

В Бейсике используются только побитовые логические операции, а операнды представляются в восьми-, шестнадцати- или тридцатидвухразрядном дополнительном коде. При этом булевым значениям False и Тruе соответствуют десятичные значения 0 и -1, так 0 — число, в котором все биты обнулены, а -1 — двоичное число, все биты которого установлены в 1 (таблица 3).

Операциям *исключающее ИЛИ* (неравнозначность), *эквивалентность* (равнозначность) и *импликация* в Бейсике соответствуют операторы XOR, EQV и IMP. Результат логической операции определяется поразрядно согласно таблице 8. Операторы приведены в порядке убывания их приоритета.

**Таблица 8 - Результаты, возвращаемые логическими операциями**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Операнды | |  |  | Результаты операций | | |  |
| Х | Y | NOT Х | Х AND Y | X OR Y | X XOR Y | X EQV Y | X IMP Y |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

***Примечание.*** В языке программирования Бейсик (и приведенных ниже заданиях) для обозначения целочисленных операндов, представленных в восьмеричной и шестнадцатеричной системах счисления, используются префиксы &O и &H соответственно.

**2.4 ЗАДАНИЯ К РАБОТЕ** Ц е л ь р а б о т ы . Изучение логических основ ЭВМ

З а д а н и е . Вычислить логические выражения. Среди предложенных

результатов (Y1 ÷ Y4) указать правильный.

**Вариант 1** *выражение 1* y = NOT ((&H23 IMP 23) AND &H5) IMP &O13

Y1=1111111111111010 Y2=101 Y3=1111 Y4=1111111111011111 *выражение 2* y = (36 EQV &H29 EQV NOT 20) AND &H16

Y1=1111111111100110 Y2=1111111111101011 Y3=110

Y4=1111111111110010 **Вариант 2** *выражение 1* y = NOT NOT (&O47 EQV &O10 EQV 17) IMP &O0

Y1=111110 Y2=1111111111000001 Y3=111110 Y4=10001 *выражение 2* y = 21 OR (&O45 EQV NOT (&O11 AND &H24)) Y1=1111111111111111 Y2=0 Y3=110101 Y4=1001 **Вариант 3** *выражение 1* y = ((&O53 XOR &H4) OR &H29) AND 17 Y1=101111 Y2=1 Y3=101111 Y4=100 *выражение 2* y = (&HD EQV &H8) OR NOT NOT (2 OR &O55)

Y1=1111111111010000 Y2=101111 Y3=101101 Y4=1111111111111111 **Вариант 4** *выражение 1* y = 15 EQV (&O55 IMP (&O32 IMP 3)) Y1=111 Y2=11 Y3=11010 Y4=101101 *выражение 2* y = (&O22 EQV (NOT &O0 IMP &H20)) OR NOT 7

Y1=1111111111111101 Y2=1111111111001101 Y3=100000 Y4=100000 **Вариант 5** *выражение 1* y = &H18 XOR &H14 EQV &H1A IMP &O14 Y1=11110 Y2=11010 Y3=1100 Y4=10100 *выражение 2* y = NOT (&O42 XOR &H1C IMP 30) OR &H17

Y1=100000 Y2=1111111111011111 Y3=110111 Y4=111110 **Вариант 6** *выражение 1* y = (&H1A OR &H23) AND &O54 AND &O21

Y1=101000 Y2=0 Y3=111011 Y4=100011

*выражение 2* y = NOT (18 EQV NOT &H29 AND &H12) XOR &HC Y1=0 Y2=1100 Y3=10010 Y4=10010 **Вариант 7** *выражение 1* y = (&HD OR &H15) AND &O7 EQV &O34

Y1=101 Y2=1111111111100110 Y3=11101 Y4=10101 *выражение 2* y = (&H8 EQV &HD) AND 25 EQV &O50

Y1=11000 Y2=11001 Y3=1111111111111010 Y4=1111111111001111 **Вариант 8** *выражение 1* y = &O25 OR &H2A OR &O12 OR 7

Y1=1111 Y2=111 Y3=1010 Y4=111111 *выражение 2* y = (&H2 OR &H2C IMP 39) OR 26

Y1=1111111111111111 Y2=100111 Y3=101110 Y4=101100 **Вариант 9** *выражение 1* y = (&H4 XOR &H25) AND &O31 OR 6 Y1=1 Y2=11001 Y3=100001 Y4=111 *выражение 2* y = (&O45 XOR &H2F) OR &H1D OR &O43 Y1=11111 Y2=111111 Y3=1010 Y4=101111 **Вариант 10** *выражение 1* y = &O54 EQV &H0 EQV &H4 IMP &H12

Y1=101000 Y2=100 Y3=1111111111010011 Y4=1111111111010111 *выражение 2* y = (&O54 EQV 43) OR &O41 OR &O13

Y1=1111111111111001 Y2=1111111111111011 Y3=111111111111100

Y4=101011

**Вариант 11** *выражение 1* y = (43 OR 6 IMP &O51) OR &O22

Y1=1111111111111001 Y2=1111111111111011 Y3=101111 Y4=110 *выражение 2* y = &H3 EQV NOT ((NOT &O6 OR &H19) AND 47)

Y1=101010 Y2=101111 Y3=1111111111111001 Y4=11001

**Вариант 12** *выражение 1* y = NOT (NOT &O16 IMP &O51 OR &H1E) IMP &H10

Y1=1111111111000000 Y2=111111 Y3=111111 Y4=111111 *выражение 2*

y = (&O44 EQV &O20) XOR &O13 IMP 14

Y1=111111 Y2=1011 Y3=1111111111001011 Y4=10000

**Вариант 13** *выражение 1* y = 46 EQV (NOT &H17 IMP NOT 31) EQV 16

Y1=100110 Y2=1111111111001001 Y3=1111111111100000 Y4=11111 *выражение 2* y = &H2A XOR (25 IMP &O51 EQV 1)

Y1=1111111111011101 Y2=1 Y3=101001 Y4=11001

**Вариант 14** *выражение 1* y = NOT NOT (&O12 OR (NOT &O7 XOR &H24)) AND 31

Y1=1111111111011110 Y2=11110 Y3=1111111111011110

Y4=1111111111011100

*выражение 2* y = 3 OR 6 EQV &HB EQV &H2A

Y1=100110 Y2=1011 Y3=111 Y4=110

**Вариант 15** *выражение 1* y= 36 AND NOT NOT (&H2A XOR 31 XOR &H1E) Y1=1111111111010100 Y2=100000 Y3=1 Y4=11110 *выражение 2* y= NOT ((&O14 XOR &H3) OR &H3) EQV NOT NOT &H14

Y1=11011 Y2=10100 Y3=1111111111110000 Y4=1111

**Вариант 16** *выражение 1* y = &H1E OR (3 EQV &O17 OR 18)

Y1=11111 Y2=10010 Y3=1111111111111111 Y4=11 *выражение 2* y = &O56 EQV &O35 XOR NOT (&H2 EQV 44)

Y1=101110 Y2=1111111111100010 Y3=101100 Y4=10

##### 2.5 ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

З а д а н и е

Вычислить логическое выражение.

Y = (38 OR &H1C) AND &H15 IMP NOT &O5

Y1=1 Y2=10100 Y3=1111111111111011 Y4=111110

Р а с ч е т з а д а н и я

Переводим все операнды в двоичную систему счисления:

38(10) = 100110(2)

1С(16) = 11100(2)

15(16) = 10101(2)

5 (8) = 101(2)

Указываем приоритет выполнения операций:

Y = (38 OR &H1C) AND &H15 IMP NOT &O5

1 2

3

4

Определяем результат выполнения каждой операции побитно, используя

для представления операндов шестнадцатиразрядный дополнительный код:

1) 38 **OR** &H1C

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

###### 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0

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

2) **NOT** &O5

###### 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

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

3) (38 OR &H1C) **AND** &H15

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

###### 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1

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

4) (38 OR &H1C) AND &H15 **IMP** NOT &O5

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

###### 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0

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

Р е з у л ь т а т в ы п о л н е н и я з а д а н и я

Y = Y3 = 1111111111111011(2) = 177773(8) = FFFB(16) = -5(10)

#### ПРИЛОЖЕНИЕ

Таблица сложения в восьмеричной системе счисления

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 11 |
| 2 | 3 | 4 | 5 | 6 | 7 | 10 | 11 | 12 |
| 3 | 4 | 5 | 6 | 7 | 10 | 11 | 12 | 13 |
| 4 | 5 | 6 | 7 | 10 | 11 | 12 | 13 | 14 |
| 5 | 6 | 7 | 10 | 11 | 12 | 13 | 14 | 15 |
| 6 | 7 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 7 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 20 |

Таблица умножения в восьмеричной системе счисления

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 |
| 2 | 2 | 4 | 6 | 10 | 12 | 14 | 16 | 20 |
| 3 | 3 | 6 | 11 | 14 | 17 | 22 | 25 | 30 |
| 4 | 4 | 10 | 14 | 20 | 24 | 30 | 34 | 40 |
| 5 | 5 | 12 | 17 | 24 | 31 | 36 | 43 | 50 |
| 6 | 6 | 14 | 22 | 30 | 36 | 44 | 52 | 60 |
| 7 | 7 | 16 | 25 | 34 | 43 | 52 | 61 | 70 |
| 10 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 100 |

Таблица сложения в шестнадцатеричной системе счисления

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 |
| 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 |
| 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 |
| 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A |
| B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B |
| C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C |
| D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E |
| F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | 20 |

Таблица умножения в шестнадцатеричной системе счисления

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| 2 | 2 | 4 | 6 | 8 | A | C | E | 10 | 12 | 14 | 16 | 18 | 1A | 1C | 1E | 20 |
| 3 | 3 | 6 | 9 | C | F | 12 | 15 | 18 | 1B | 1E | 21 | 24 | 27 | 2A | 2D | 30 |
| 4 | 4 | 8 | C | 10 | 14 | 18 | 1C | 20 | 24 | 28 | 2C | 30 | 34 | 38 | 3C | 40 |
| 5 | 5 | A | F | 14 | 19 | 1E | 23 | 28 | 2D | 32 | 37 | 3C | 41 | 46 | 4B | 50 |
| 6 | 6 | C | 12 | 18 | 1E | 24 | 2A | 30 | 36 | 3C | 42 | 48 | 4E | 54 | 5A | 60 |
| 7 | 7 | E | 15 | 1C | 23 | 2A | 31 | 38 | 3F | 46 | 4D | 54 | 5B | 62 | 69 | 70 |
| 8 | 8 | 10 | 18 | 20 | 28 | 30 | 38 | 40 | 48 | 50 | 58 | 60 | 68 | 70 | 78 | 80 |
| 9 | 9 | 12 | 1B | 24 | 2D | 36 | 3F | 48 | 51 | 5A | 63 | 6C | 75 | 7E | 87 | 90 |
| A | A | 14 | 1E | 28 | 32 | 3C | 46 | 50 | 5A | 64 | 6E | 78 | 82 | 8C | 96 | A0 |
| B | B | 16 | 21 | 2C | 37 | 42 | 4D | 58 | 63 | 6E | 79 | 84 | 8F | 9A | A5 | B0 |
| C | C | 18 | 24 | 30 | 3C | 48 | 54 | 60 | 6C | 78 | 84 | 90 | 9C | A8 | B4 | C0 |
| D | D | 1A | 27 | 34 | 41 | 4E | 5B | 68 | 75 | 82 | 8F | 9C | A9 | B6 | C3 | D0 |
| E | E | 1C | 2A | 38 | 46 | 54 | 62 | 70 | 7E | 8C | 9A | A8 | B6 | C4 | D2 | E0 |
| F | F | 1E | 2D | 3C | 4B | 5A | 69 | 78 | 87 | 96 | A5 | B4 | C3 | D2 | E1 | F0 |
| 10 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | A0 | B0 | C0 | D0 | E0 | F0 | 100 |

**Базовая таблица кодировки ASCII**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 beep (звуковой сигнал) | 1. % 2. & | 1. 4 2. 5 | 67  68 | C D | 82  83 | R S | 97  98 | a b | 112  113 | p q |
| 8 backspace  (удаление предыдущего сим-  вола) | 1. ' 2. ( 3. ) | 1. 6 2. 7 3. 8 | 69  70  71 | E  F  G | 84  85  86 | T  U  V | 99  100  101 | c d e | 114  115  116 | r s t |
| 9 tab (табуляция) | 42 \* | 57 9 | 72 | H | 87 | W | 102 | f | 117 | u |
| 10 linefeed (перевод строки) | 1. + 2. , | 1. : 2. ; | 73  74 | I J | 88  89 | X  Y | 103  104 | g h | 118  119 | v  w |
| 13 carriage  return (возврат каретки) | 45 - 46 . 47 / | 1. < 2. = 3. > | 75  76  77 | K L  M | 90  91  92 | Z  [  \ | 105  106  107 | i j  k | 120  121  122 | x y z |
| 32 space (пробел) |
| 33 ! | 48 0 | 63 ? | 78 | N | 93 | ] | 108 | l | 123 | { |
| 34 " | 49 1 | 64 @ | 79 | O | 94 | ^ | 109 | m | 124 | | |
| 35 # | 50 2 | 65 A | 80 | P | 95 | \_ | 110 | n | 125 | } |
| 36 $ | 51 3 | 66 B | 81 | Q | 96 | ` | 111 | o | 126 | ~ |