

**II. Внешняя память** реализована с помощью специальных устройств, которые называются *накопителями*. **Накопители** состоят из двух частей: *носителя* (устройства, на котором хранится информация) и *привода* (устройства, предназначенного для считывания информации из носителя и записи информации на носитель).

### Виды накопителей

**1. Магнитные** (магнитная запись основана на представлении цифровой информации в виде нулей и единиц, преобразованной в переменный электрический ток, который сопровождается переменным магнитным полем; немагнитным участкам соответствует 0, намагнитным – 1):

**а) жесткий магнитный диск (HDD – Hard Disk Drive)** (рис.1.32). В 1973 году фирма IBM выпустила первый жесткий магнитный диск, имеющий 30 дорожек по 30 секторов. По аналогии с винтовками, имеющими калибр 30/30, его назвали *винчестером*. *Винчестер* представляет собой несколько керамических или алюминиевых пластин, на которые наносится специальный магнитный слой;



Рис.1.32. Жесткий магнитный диск



Рис.1.33. Гибкий магнитный диск

**б) гибкий магнитный диск (FDD – Floppy Disk Drive)** (рис.1.33) – это тонкая пластиковая основа с нанесенным магнитным шаром.

**2. Оптические диски** (рис. 1.34). Закодированная информация наносится на диск лазерным лучом, который образует на его поверхности микроскопические углубления, разделенные плоскими участками. Цифровая информация подается с помощью чередования углублений (*неотражающих* пятен) и островков, *отражающих* свет. Считывание происходит при помощи лазерного луча, который, попадая на островок, отражающий свет, отклоняется на *фотодетектор*, который интерпретирует его как двоичную единицу; если попадает в углубление, то рассеивается и поглощается, фотодетектор фиксирует двоичный ноль.

**CD-ROM** – устройство только для считывания информации;

**CD-R** – устройство для однократной записи информации;

**CD-RW** – устройство для многократной записи информации;

**DVD-ROM, DVD-R, DVD-RW** – устройства для двусторонней записи.



Рис.1.34. Дисковод CD/DVD



Рис.1.35. USB флеш-диск

**3. Магнитооптические** совмещают технологии магнитных и оптических устройств.

**4. Стримеры** – это устройства для записи на магнитную ленту.

**5. Флеш-память** - это энергонезависимое полупроводниковое запоминающее устройство, выполненное в виде микросхемы (*чипа*), в которую можно записывать данные и хранить их как угодно долго. Стирание производится электрическим разрядом, после чего можно записывать новые данные непосредственно в компьютере.

**Флеш-диск (*flash disc*), USB-флеш-память, USB-память** (рис.1.35) конструктивно представляет собой "брелок" продолговатой формы, состоящий из защитного колпачка и собственно накопителя с **USB-разъемом** (одна или две микросхемы флеш-памяти и USB-контроллер). *Достоинствами* накопителей этого вида являются:

а) возможность подключения их непосредственно к компьютеру для переноса данных без помощи других устройств;

б) значительный объем памяти (от 32 Мбайт до 16 Гбайт).

### 1.1.3. Кодирование информации. Системы счисления

Компьютер хранит и обрабатывает информацию в виде **двоичных кодов** – последовательности цифр 0 и 1 (**битов**). Научимся представлять любые числа в виде комбинации нулей и единиц, то есть *переводить числа из десятичной системы счисления в двоичную*.

**Система счисления** – это система записи чисел с помощью определенного набора знаков.

Мы пользуемся **десятичной системой счисления**. Она использует 10 цифр: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0**. В этой системе любое целое неотрицательное число можно представить в виде суммы степеней числа 10.

*Например:*  $7586 = 7 \cdot 10^3 + 5 \cdot 10^2 + 8 \cdot 10^1 + 6 \cdot 10^0$ .

Число **10** называется **основанием системы счисления**, числа 7, 5, 8, 6 – **коэффициентами**.

Представим любое число (*например*, 25) в виде степеней числа 2:  $25 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ . Здесь **основанием** является число 2, а 1, 1, 0, 0, 1 – это **коэффициенты**.

**Двоичная система счисления** использует две цифры: **0** и **1**, ее **основанием** является число **2**. *Например:*  $10100101_2, 11111000_2$  – числа, записанные в двоичной системе счисления.

**Задание 1.** Переведите числа 37 и 197 из десятичной системы счисления в двоичную ( $37_{10} \rightarrow x_2$  -?  $197_{10} \rightarrow x_2$  -?).

**Решение**

$37 : 2 = 18$ (остаток $\boxed{1}$ ); $18 : 2 = 9$ (остаток $\boxed{0}$ ); $9 : 2 = 4$ (остаток $\boxed{1}$ ); $4 : 2 = 2$ (остаток $\boxed{0}$ ); $2 : 2 = \boxed{1}$ (остаток $\boxed{0}$ ).	$197 : 2 = 98$ (остаток $\boxed{1}$ ); $98 : 2 = 49$ (остаток $\boxed{0}$ ); $49 : 2 = 24$ (остаток $\boxed{1}$ ); $24 : 2 = 12$ (остаток $\boxed{0}$ ); $12 : 2 = 6$ (остаток $\boxed{0}$ ); $6 : 2 = 3$ (остаток $\boxed{0}$ ); $3 : 2 = \boxed{1}$ (остаток $\boxed{1}$ ).
--	---

**Ответ:**  $37_{10} = 100101_2$ .

**Ответ:**  $197_{10} = 11000101_2$ .

**Задание 2.** Переведите числа  $111000_2$  и  $10011000_2$  из двоичной системы счисления в десятичную ( $111000_2 \rightarrow x_{10}$  -?  $10011000_2 \rightarrow x_{10}$  -?).

**Решение**

$$111000_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 120_{10}.$$

$$10011000_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 152_{10}.$$

**Ответ:**  $111000_2 = 120_{10}$ ;  $10011000_2 = 152_{10}$ .

Обратите внимание, что  $\boxed{2^0=1}$ .

При выполнении **арифметических действий в двоичной системе счисления** выполняются правила:

**Правило сложения:**  $0+0=0$ ;  $0+1=1$ ;  $1+0=1$ ;  $1+1=10$ .

**Правило вычитания:**  $0-0=0$ ;  $1-0=1$ ;  $1-1=0$ ;  $10-1=1$ .

**Правило умножения:**  $0 \cdot 0=0$ ;  $0 \cdot 1=0$ ;  $1 \cdot 0=0$ ;  $1 \cdot 1=1$ .

**Задание 3.** Выполните сложение, вычитание, умножение и деление чисел 15 и 5 в двоичной системе счисления.

**Решение**

$$15_{10} = 1111_2, 5_{10} = 101_2.$$

$$\begin{array}{r} 1111 \\ + 101 \\ \hline 10100 \end{array} = 20_{10}$$

$$\begin{array}{r} 1111 \\ - 101 \\ \hline 1010 \end{array} = 10_{10}$$

$$\begin{array}{r} 1111 \\ \times 101 \\ + 1111 \\ \hline 1111 \\ \hline 1001011 \end{array} = 75_{10}$$

$$1111 : 101 = 11_2 = 3_{10}$$

**Шестнадцатеричная система счисления** использует 15 цифр: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** и буквы: **A = 10<sub>10</sub>, B = 11<sub>10</sub>, C = 12<sub>10</sub>, D = 13<sub>10</sub>, E = 14<sub>10</sub>, F = 15<sub>10</sub>**.

**Задание 4.** Переведите число 317 из *десятичной* системы счисления в *шестнадцатеричную* (317<sub>10</sub> → x<sub>16</sub> -?).

**Решение**

$$317 : 16 = 19 \text{ (ост. } \boxed{13} \text{)};$$

$$19 : 16 = \boxed{1} \text{ (ост. } \boxed{3} \text{)}.$$

**Ответ:** 317<sub>10</sub> = 13D<sub>16</sub>.

**Задание 5.** Переведите число 110<sub>16</sub> из *шестнадцатеричной* системы счисления в *десятичную* (110<sub>16</sub> → x<sub>10</sub> -?).

**Решение**

$$1 \cdot 16^2 + 1 \cdot 16^1 + 0 \cdot 16^0 = 272_{10}. \text{ **Ответ:}** } 110_{16} = 272_{10}.$$

**Задание 6.** Переведите число 14A<sub>16</sub> из *шестнадцатеричной* системы счисления в *двоичную* (14A<sub>16</sub> → x<sub>2</sub> -?).

**Решение**

Нужно выполнить действия: 14A<sub>16</sub> → y<sub>10</sub> → x<sub>2</sub> -?

1)  $1 \cdot 16^2 + 4 \cdot 16^1 + 10 \cdot 16^0 = 256 + 64 + 10 = 330_{10}$ ;

2)  $330 : 2 = 165 \text{ (остаток } \boxed{0} \text{)};$   
 $165 : 2 = 82 \text{ (остаток } \boxed{1} \text{)};$   
 $82 : 2 = 41 \text{ (остаток } \boxed{0} \text{)};$   
 $41 : 2 = 20 \text{ (остаток } \boxed{1} \text{)};$   
 $20 : 2 = 10 \text{ (остаток } \boxed{0} \text{)};$   
 $10 : 2 = 5 \text{ (остаток } \boxed{0} \text{)};$   
 $5 : 2 = 2 \text{ (остаток } \boxed{1} \text{)};$   
 $2 : 2 = \boxed{1} \text{ (остаток } \boxed{0} \text{)};$

**Ответ:** 14A<sub>16</sub> = 101001010<sub>2</sub>.

Можно выполнить **Задание 6** проще, то есть чтобы перевести *шестнадцатеричное* число в *двоичную* систему счисления, нужно каждую *шестнадцатеричную* цифру этого

числа заменить соответствующим ему четырехзначным двоичным кодом – **тетрадой**. Для этого используют кодовую таблицу:

шестнадцатеричные цифры	тетрады	шестнадцатеричные цифры	тетрады
0	0000	8	1000
1	0001	9	1001
2	0010	10 = A	1010
3	0011	11 = B	1011
4	0100	12 = C	1100
5	0101	13 = D	1101
6	0110	14 = E	1110
7	0111	15 = F	1111

В нашем примере  $14A_{16} = \underbrace{0001}_{1} \underbrace{0100}_{4} \underbrace{1010}_{A=10}_2 = 101001010_2$ .

Наоборот, чтобы перевести двоичное число в шестнадцатеричную систему счисления, нужно разбить это число справа налево на тетрады и заменить каждую тетраду соответствующей шестнадцатеричной цифрой.

**Задание 7.** Переведите число  $1011110001_2$  из двоичной системы счисления в шестнадцатеричную ( $1011110001_2 \rightarrow x_{16} - ?$ ).

**Решение**

$$1011110001_2 = \underbrace{0010}_2 \underbrace{1111}_F \underbrace{0001}_1_2 = 101001010_2$$

**Ответ:**  $1011110001_2 = 2F1_{16}$ .

Простота перехода из шестнадцатеричной системы счисления в двоичную и наоборот послужила использованию шестнадцатеричной системы счисления для сокращенной записи двоичных чисел. Команды программ для компьютера записываются на бумаге в шестнадцатеричной системе счисления, а при вводе их в компьютер автоматически преобразовываются в двоичные коды.

Вычислим, *какое наибольшее двоичное число можно записать, используя заданное количество битов, а так же сколько всего существует таких чисел.*

**3 бита:** наибольшее число  $111_2 = 7_{10} = 2^3 - 1$ , всего существует  $2^3 = 8$  чисел (000, 001, 010, 011, 100, 101, 110, 111);

**4 бита:** наибольшее число  $1111_2 = 15_{10} = 2^4 - 1$ , всего существует  $2^4$  чисел (по аналогии с предыдущим);

...

**n битов:** наибольшее число  $2^n - 1$ , всего существует  $2^n$  чисел.

*Значит, с помощью n битов можно записать двоичные коды чисел от 0 до  $2^n - 1$ , то есть всего  $2^n$  чисел.*

**Пример.** Сколько чисел в десятичном виде можно записать, используя 1 байт (8 битов)?

**Ответ.**  $2^8 = 256$  чисел. Это числа от 0 до 255.

Использование шестнадцатеричной системы в информатике является очень удобным, так как содержимое одного байта можно записать двумя шестнадцатеричными цифрами. Наибольшая шестнадцатеричная цифра  $F_{16} = 15_{10} = 1111_2$ . Значит, для записи любой шестнадцатеричной цифры в двоичном виде требуется не более четырех битов.

**Например:**  $33_{10} = 21_{16} = 00100001_2$  ( $0010_2 = 2_{16}$ ,  $0001_2 = 1_{16}$ ).

$168_{10} = A8_{16} = 10101000_2$  ( $1010_2 = A_{16}$ ,  $1000_2 = 8_{16}$ ).

$255_{10} = FF_{16} = 11111111_2$  ( $1111_2 = F_{16}$ ).

Шестнадцатеричная система используется, в частности, для кодирования адресов ячеек оперативной памяти.

**Пример.** Адрес ячейки оперативной памяти такой: FA21. Определите ее номер. **Решение:**

$(FA21)_{16} = 15 \cdot 16^3 + 10 \cdot 16^2 + 2 \cdot 16 + 1 = 64033$ .

**Ответ.** Номер ячейки 64033.

Ознакомимся с **таблицей компьютерных кодов ASCII** (*American Standard Code for Information Interchange* –

*американский стандартный код для обмена информацией*). Мы уже выяснили, что всего есть 256 разных байтов. Их нумеруют цифрами от 0 до 255:

0	00000000
1	00000001
2	00000010
...	
50	00011010
...	
255	11111111

Байты с номерами *от 0 до 31* используются для кодирования служебных символов, которые не используются в текстовых документах.

Байты с номерами *от 32 до 127* используются для кодирования цифр, букв английского алфавита и служебных символов, которые используются в текстовых документах.

Байты с номерами *от 128 до 255* используются для национальных алфавитов и символов псевдографики.

*Например:*

<i>Номер кода</i>	<i>Символ</i>	<i>Номер кода</i>	<i>Символ</i>
32	пропуск	58	:
33	!	64	@
43	+	65	A
47	/	66	B
48	0	67	C
49	1	90	Z
57	9	97	a

Информация кодируется автоматически, запоминать коды не нужно.

Для кодирования одного символа *алфавитно-цифровой* информации используется *1 байт*.

### **Кодирование графической информации**

Оттенки серого цвета от черного до белого автоматически градуируются на 256 значений и кодируются байтами. Любое черно-белое изображение – это упорядоченный набор точек.

Каждой точке ставится в соответствие байт, который означает оттенок серого цвета. Если изображение *цветное*, то одной точке ставится в соответствие *три байта*. Любой цвет является результатом некоторого смешивания трех основных цветов: *красного (red)*, *зеленого (green)* и *синего (blue)*. Каждый из них имеет яркость с 256-значной градацией – от 0 до 255. В компьютерной графике такую палитру называют **RGB-палитрой**, где красный цвет имеет код (255, 0, 0), зеленый – (0, 255, 0), синий – (0, 0, 255). Значит, всего разных цветов существует  $256 \cdot 256 \cdot 256$ , то есть больше 16 миллионов.

### Кодирование звука

Звук характеризуется силой и длительностью. Силу звука градуируют на 256 значений, в каждый момент времени некоторому звуку соответствует значение некоторого байта. Это называют *оцифровыванием звука*.

#### ВОПРОСЫ

1. Что такое информатика?
2. Что такое информация?
3. Какие существуют виды информации?
4. Что такое входная информация?
5. Что такое выходная информация?
6. Что такое внутренняя информация?
7. Какие вы знаете информационные процессы?
8. Какие есть формы подачи информации?
9. Какие вы знаете способы подачи информации?
10. Что такое носитель информации?
11. Кто может быть носителем информации?
12. Что такое шум?
13. Какие есть свойства информации?
14. Что такое 1 бит?
15. Какие существуют единицы измерения информации?
16. Что такое компьютер?
17. Какие вы знаете основные составляющие компьютера? Что такое процессор?
18. Что такое запоминающие устройства?
19. Что такое аппаратная часть компьютера?

20. Что такое программная часть компьютера?
21. Из чего состоит процессор?
22. Какие вы знаете основные характеристики процессора?
23. Что такое сопроцессор?
24. Какие вы знаете устройства ввода информации?
25. Какие есть устройства вывода информации?
26. Что является основной характеристикой дисплея?
27. Какие существуют режимы работы дисплея?
28. Что такое контроллер?
29. Что такое видеоадаптер?
30. Какие есть виды принтеров?
31. Что такое модем? Какая его основная характеристика?
32. Какие существуют виды памяти компьютера?
33. Какие есть виды внутренней памяти?
34. Что такое разрядность?
35. Какие существуют виды внешней памяти?
36. Какие вы знаете виды магнитных накопителей?
37. Какие есть виды оптических накопителей?
38. Что такое 1 бод?
39. Как наносится информация на магнитные и оптические носители?
40. Что такое система счисления?
41. Какие вы знаете системы счисления?
42. Что такое основание системы счисления?
43. Сколько десятичных чисел в двоичном виде можно записать, используя заданное количество битов?
44. Сколько есть разных байтов?
45. Как кодируется информация с помощью таблицы ASCII-кодов?
46. Сколько битов нужно для кодирования одного символа алфавита?
47. Как кодируется графическая информация?
48. Как кодируется звуковая информация?

#### **ЗАДАНИЯ**

1. Переведите числа из *десятичной* системы счисления в *двоичную*:
 

а) $34_{10}$ и $179_{10}$ ;	в) $99_{10}$ и $112_{10}$ ;
б) $156_{10}$ и $85_{10}$ ;	г) $23_{10}$ и $103_{10}$ .

2. Переведите числа в *шестнадцатеричную* систему счисления:  
а)  $89_{10}$ ; б)  $145_{10}$ ; в)  $211_{10}$ ; г)  $100_{10}$ .
3. Переведите числа в *двоичную* и *шестнадцатеричную* системы счисления: а)  $39_{10}$ ; б)  $165_{10}$ ; в)  $158_{10}$ ; г)  $100_{10}$ .
4. Переведите числа из *двоичной* системы счисления в *десятичную*:  
 |  $1011_2$  и  $10001_2$ ; |  $110111_2$  и  $100101_2$ ;  
 |  $11011_2$  и  $100000_2$ ; |  $1101011_2$  и  $11111001_2$ .
5. Переведите числа из *шестнадцатеричной* системы в *десятичную*: а)  $23B_{16}$ ; б)  $1C3_{16}$ ; в)  $1DF_{16}$ ; г)  $ABC7_{16}$ .
6. Переведите числа из *двоичной* системы в *шестнадцатеричную*:  
 |  $1000_2$  и  $1011100_2$ ; |  $101100_2$  и  $10100100_2$ ;  
 |  $100110_2$  и  $111000_2$ ; |  $10001100_2$  и  $100111000_2$ .
7. Вычислите сумму и разность чисел в *двоичной* системе счисления:  
 а)  $37_{10}$  и  $168_{10}$ ; | в)  $99_{10}$  и  $102_{10}$ ;  
 б)  $154_{10}$  и  $85_{10}$ ; | г)  $123_{10}$  и  $79_{10}$ .
8. Вычислите произведение и частное чисел в *двоичной* системе счисления:  
 а)  $39_{10}$  и  $145_{10}$ ; | в)  $99_{10}$  и  $104_{10}$ ;  
 б)  $197_{10}$  и  $95_{10}$ ; | г)  $127_{10}$  и  $93_{10}$ .
9. Вычислите, сколько байт содержит:  
 а) 2 Мегабайта; | в) 1 Гигабайт;  
 б) 3 килобайта; | г) 5 Гигабайт.
10. Вычислите, сколько битов содержит:  
 а) 3 Мегабайта; | б) 1 Гигабайт.

## 1.2. Компьютерная арифметика и компьютерная логика

### *Новые слова*

*Русский язык*

арифметический

база

быстродействие

*English*

arithmetical

base

quick-action

<b>возведение</b>	raising
<b>вычитание</b>	subtraction
<b>данные</b>	dates
<b>деление</b>	division
<b>диапазон</b>	diapason
<b>дополнять</b>	supply
<b>знак</b>	sign
<b>импульс</b>	impulse
<b>инвертировать</b>	invert
<b>интегрированный</b>	integrate
<b>истина</b>	true
<b>логика</b>	logic
<b>ложь</b>	false
<b>машина</b>	machine
<b>миллион</b>	million
<b>модель</b>	model
<b>напряжение</b>	voltage
<b>операция</b>	operation
<b>определить</b>	determine
<b>отрицание</b>	negation
<b>переполнение</b>	overflow
<b>полупроводник</b>	semiconductor
<b>прибавить</b>	add
<b>признак</b>	sign
<b>приоритет</b>	priority
<b>программирование</b>	programming
<b>противоположный</b>	opposite
<b>пусть</b>	let
<b>распознавать</b>	perceive
<b>регистр</b>	register
<b>резистор</b>	resistor
<b>смещение</b>	displacement
<b>состояние</b>	state
<b>способность</b>	quality
<b>стадион</b>	stadium
<b>степень</b>	factor
<b>суждение</b>	argument, proposition
<b>таблица</b>	table

точность	accuracy
транзистор	transistor
триггер	trigger
умножение	multiplication
установить	establish
утверждение	approval, statement
формат	format
функция	function
цель	goal
элемент	element

### 1.2.1. Компьютерная арифметика

Информация в компьютере *представлена, передается и фиксируется* с помощью **импульсных электрических сигналов**. Сначала она автоматически кодируется последовательностью нулей и единиц. В электронных устройствах коду **1** соответствует **наличие** электрического импульса, а коду **0** – **отсутствие** импульса (*пауза*). Закодированная информация хранится в регистрах процессора (на протяжении одного такта), или в оперативной памяти (во время выполнения программы), или в постоянной или дисковой памяти (здесь она хранится постоянно).

Важными элементами процессора являются **регистры**. Один регистр обрабатывает 16, 32 бита информации или больше. Это число определяет **разрядность** регистра и процессора в целом. В *персональных компьютерах* используют в основном 16- и 32-разрядные микропроцессоры, в более совершенных моделях и в больших компьютерах – 64-разрядные.

Регистр составляют **триггеры** – устройства, хранящие 1 бит информации: 0 или 1. Триггер может находиться в одном из двух устойчивых состояний, пока работает компьютер и пока на входы триггера не поступает сигнал изменения состояния. Одно из состояний обозначается “1”, и ему соответствует **наличие напряжения** (это входной сигнал) на главном выходе триггера. Если на **вход** триггера подать *сигнал изменения состояния*, то триггер перейдет в

противоположное состояние, например, состояние “0”, при котором **напряжения** на главном **выходе нет**. Один триггер базируется на двух транзисторах.

Транзисторы, резисторы и другие элементы настолько малы, что миллионы их можно разместить на небольших *полупроводниковых пластинках*. Это и есть **интегрированные устройства**.

Для *функционирования* триггера необходим **электрический ток**. Поэтому после выключения питания информация в оперативной памяти исчезает. Постоянная память устроена так, что информация в ней остается даже после выключения тока.

**Высокое быстроедействие** компьютера достигается благодаря свойству элементов *менять состояния* миллионы раз в секунду. Это свойство определяется **тактовой частотой**, которая зависит от физических свойств элементов электронной схемы.

*Большая точность компьютера основывается на способности машины однозначно распознавать, проходят ли импульсы (до 5В) в электрических цепях, или есть паузы(напряжение меньше 0,5В).*

Рассмотрим 8-разрядный регистр процессора и **действия с целыми числами** в диапазоне, *например*, от -128 до 127. Сначала числа кодируются двоичными кодами с использованием *дополняющей арифметики*. **Дополняющая арифметика** – это способ кодирования, который не меняет двоичные коды положительных чисел, а отрицательное число кодирует некоторым двоичным числом без знака. Это дает возможность **вычитания** двух чисел **заменить** действием **сложения** их *дополняющих кодов*. **Дополняющий двоичный код отрицательного числа** образуют так: все разряды кода соответствующего *положительного* числа **инвертируют** и **прибавляют** к коду единицу. Например, рассмотрим дополняющие коды для чисел из диапазона -128...127:

Число	Дополняющий код	Число	Дополняющий код
0	00000000	-128	10000000
1	00000001	-127	10000001
2	00000010	-126	10000010
3	00000011	...	
...		-2	11111110
127	01111111	-1	11111111

**Сложение двоичных кодов** выполняется по обычным правилам. Так как разрядов есть всего 8, то единица, которая должна была переноситься дальше после сложения цифр в старших разрядах, теряется, но оказывается, что именно это дает нужный результат.

**Например:**

$3 - 1 = 3 + (-1) = (00000011)_2 + (11111111)_2 = (00000010)_2 = 2$ , так как единица утеряна здесь:

$$\begin{array}{r} 00000011 \\ + \quad 11111111 \\ \hline \rightarrow 00000010. \end{array}$$

Рассмотрим двоичные коды двух пар чисел: 2 и 4, 3 и 6 и выясним, как процессор выполняет умножение и деление некоторого целого числа на 2:

2	00000010	3	00000011
4	00000100	6	00000110

**Умножение числа на 2 процессор выполняет путем сдвига значений регистров влево, а деления – сдвига регистров вправо.**

**Пример.** Умножение  $4 \cdot 3 = 4 \cdot 2 + 4$  процессор выполняет так: в числе 00000100 (это есть 4) сдвигает единицу влево и получает 00001000, прибавляет 00000100:

$$00001000 + 00000100 = 000001100 = 12.$$

**Второй способ** выполнения операции умножения целых чисел заключается в замене ее операцией сложения:  $4 \cdot 3 = 4 + 4 + 4$ .

Рассмотрим понятие **переполнения**. Пусть нужно к числу 120 прибавить 9. На 8-разрядном регистре нельзя выполнить такое сложение, так как результат выходит за границы

допустимого диапазона (-128...127). В таком случае говорят, что произошло *переполнение* – результаты будут неправильными, а система сообщит об ошибке.

**Пример.** Рассмотрим возникновение переполнения для операции сложения двух положительных чисел: 120 и 9.

$$\begin{array}{r} 120 \longrightarrow 01111000 \\ + \\ \underline{\phantom{0}9} \longrightarrow \underline{00001001} \\ -127 \longleftarrow 10000001 \end{array}$$

Вместо правильного результата 129 получили -127. Как процессор определил, что произошло переполнение? Нули в *старших разрядах* дополняющих кодов чисел 120 и 9 – это признаки положительных чисел. Результат отрицательным быть не может. В приведенном примере единица в старшем разряде результата – это признак отрицательного числа. Так как во время сложения положительных чисел 120 и 9 в 8-разрядном регистре процессора появилась единица, то это значит, что произошла *ошибка*, которая и называется *переполнением*. Чтобы сложение больших чисел выполнялось правильно, надо или большее количество 8-разрядных регистров, или один многоразрядный регистр: 16-или 32-разрядный (именно для этого в языках программирования служат команды объявления типов данных, например, *byte* или *integer* в языке Паскаль).

### 1.2.2. Компьютерная логика. Логические операции

Решая задачи по математике, физике, химии, мы *оперируем данными*. Этими данными являются числа. Мы умеем выполнять действия с числовыми данными: *сложение, вычитание, умножение, деление, возведение в степень*. Но в информатике существуют и другие типы данных, например, *логические*. Компьютер обрабатывает двоичные коды с помощью *арифметических и логических операций*.

Все элементы, узлы, блоки и устройства, из которых состоит компьютер, являются преобразователями информации – *цифровыми автоматами*.

**Цифровой автомат** – это устройство, предназначенное для преобразования *дискретной (цифровой)* информации и способное переходить под действием *входных* сигналов из одного состояния в другое и выдавать *выходные* сигналы.

Функционирование цифровых автоматов (*сумматоров, шифраторов* и других) описываются формулами *булевой алгебры*. Ее создал английский математик Джордж Буль (1815-1864). *Булева алгебра* – это алгебра, в которой используются логические константы «0» и «1» и логические операции над ними: *логическое отрицание, логическое сложение, логическое умножение*.

Цифровой автомат строится из *логических элементов*. **Логические элементы** – это элементы, которые реализуют основные логические операции. Из этих элементов строят *функциональные схемы*, которые соответствуют некоторым структурным формулам. *Структурная формула* – это формула, которая задает некоторую булеву функцию, которая описывает логику работы цифрового автомата.

Мы научились выполнять арифметические действия с двоичными числами. Есть еще один вид действий с информацией, которым мы пользуемся в повседневной жизни: *определяем, какой является информация: истинной или ложной*.

**I. Например**, мы говорим: “Я пойду на стадион **или** в парк”.

Если мы **не пошли** на стадион, и **не пошли** в парк, то это утверждение является **ложным**.

Если мы **не пошли** на стадион, но **пошли** в парк, то утверждение является **истинным**.

Если мы **пошли** на стадион, но **не пошли** в парк, то утверждение является **истинным**.

Если мы **пошли** на стадион и **пошли** в парк, то утверждение является **истинным**.

Можно представить эти суждения в виде такой **таблицы истинности**:

<b>“Я пойду на стадион”</b>	<b>“Я пойду в парк”</b>	<b>“Я пойду на стадион или в парк”</b>
Ложь	Ложь	Ложь
Ложь	Истина	Истина
Истина	Ложь	Истина
Истина	Истина	Истина

Обозначим *истину* цифрой **1**, *ложь* – цифрой **0**, утверждение **“Я пойду на стадион”** - буквой *A*, утверждение **“Я пойду в парк”** - буквой *B*, **“или”** - знаком “ + ”, тогда таблица примет вид:

<i>A</i>	<i>B</i>	<i>C = A + B</i>
0	0	0
0	1	1
1	0	1
1	1	1

Операцию **“или”** называют **логическим сложением**.

**Свойство операции «или»:** полученная с помощью этой операции новая информация истинна, если *хотя бы одна* из двух частей, принимающих участие в операции, является истинной.

**II.** Пусть мы говорим: **“Я пойду на стадион и в парк”**.

Если мы **не пошли** на стадион и **не пошли** в парк, то это утверждение является **ложным**.

Если мы **не пошли** на стадион, но **пошли** в парк, то утверждение является **ложным**.

Если мы **пошли** на стадион, но **не пошли** в парк, то утверждение является **ложным**.

Если мы **пошли** на стадион и **пошли** в парк, то утверждение является **истинным**.

Составим **таблицу истинности**:

<b>“Я пойду на стадион”</b>	<b>“Я пойду в парк”</b>	<b>“Я пойду на стадион и в парк”</b>
Ложь	Ложь	Ложь
Ложь	истина	Ложь
Истина	Ложь	Ложь
Истина	Истина	Истина

Если обозначить слово “*и*” знаком “ $*$ ”, то таблица примет вид:

$A$	$B$	$C = A * B$
0	0	0
0	1	0
1	0	0
1	1	1

Операцию “*и*” называют **логическим умножением**.

**Свойство операции «и»:** полученная с помощью этой операции новая информация истинна, если **обе части**, принимающие в ней участие, истинны.

**III.** Пусть мы говорим: “*Я не пойду в парк*”.

Если мы **не пошли в парк**, это утверждение является **истинным**.

Если мы **пошли в парк**, утверждение является **ложным**.

*Таблица истинности:*

“ <i>Я пойду на стадион</i> ”	“ <i>Я не пойду на стадион</i> ”
Истина	Ложь
Ложь	Истина

Если обозначить “*не A*” знаком “ $\bar{A}$ ”, то таблица примет вид:

$A$	$B = \bar{A}$
0	1
1	0

Операцию «*Не*» называют **логическим отрицанием**.

Все эти функции называют **логическими операциями**.

### Приоритет выполнения операций

1. Логическое отрицание.
2. Логическое умножение.
3. Логическое сложение.

**Пример.** Составьте таблицы истинности для выражений:

а)  $C = A + \overline{B * A}$ ;

### Решение

$A$	$B$	$B * A$	$\overline{B * A}$	$C = A + \overline{B * A}$
1	0	0	1	1
0	1	0	1	1
0	0	0	1	1
1	1	1	0	1

$$\text{б) } C = \overline{(B + A)} * B.$$

**Решение:**

$A$	$B$	$B+A$	$\overline{B+A}$	$C = \overline{(B+A)} * B$
1	0	1	0	0
0	1	1	0	0
0	0	0	1	0
1	1	1	0	0

Так как компьютер использует данные, представленные в двоичной системе счисления, информация об истинности или ложности должна быть представлена двоичным числом. Для сохранения информации об истинности или ложности информации достаточно одного бита. Значение этого бита равно **1**, если информация *истинна*, и равно **0**, если информация *ложна*.

### ВОПРОСЫ

1. Как представлена информация в компьютере?
2. Что такое триггер?
3. Что такое интегрированное устройство?
4. Что такое дополняющая арифметика?
5. Что такое переполнение?
6. Как процессор умножает число на 2?
7. Как объяснить операцию  $1+1=10$  в двоичной арифметике?
8. Какие вы знаете логические операции?
9. Как называют операцию „или“?
10. Как называют операцию „и“?
11. Как называют операцию „не“?
12. Как представлена в компьютере информация об истинности и ложности?

### ЗАДАНИЯ

Составьте таблицы истинности для выражений:

$$1. C = B + \overline{A} * B;$$

$$6. C = A * \overline{(B + A)} + \overline{B} * A;$$

$$2. C = A * B + \overline{A};$$

$$7. C = \overline{(A + B)} * (\overline{A} + B);$$

$$3. C = A + \overline{B} * A;$$

$$8. C = B + A * \overline{\overline{A + B}} + A * \overline{B};$$

$$4. C = B + \overline{B * A + A};$$

$$9. C = A * \overline{B} + (A + A * B);$$

$$5. C = A * A + \overline{B} * B;$$

$$\begin{array}{ll}
10. C = \overline{\overline{A}} + A * \overline{\overline{A+B}}; & 15. C = (A + B * \overline{\overline{A}}) * \overline{\overline{A+B}}; \\
11. C = \overline{\overline{B * A + A + B * A}}; & 16. C = \overline{\overline{B + B * A + \overline{\overline{A+B}}}}; \\
12. C = \overline{\overline{A + \overline{\overline{A * B + A * B}}}}; & 17. C = \overline{\overline{A + B * \overline{\overline{A * B + A * B}}}}; \\
13. C = \overline{\overline{B + \overline{\overline{A * A + \overline{\overline{B}}}}}}; & 18. C = \overline{\overline{B + A + A * \overline{\overline{B + \overline{\overline{A+B * A}}}}}}; \\
14. C = \overline{\overline{A * A + B * \overline{\overline{A+B}}}}; & 19. C = \overline{\overline{(A+B) * A + B * \overline{\overline{A+B}}}}.
\end{array}$$

### ЗАДАНИЯ К КОНТРОЛЬНОЙ РАБОТЕ №1

Контрольная работа состоит из 5 заданий, в первом из которых нужно перевести число из десятичной системы счисления в двоичную, во втором – перевести число из двоичной системы в десятичную, в третьем – перевести число из одной системы счисления в другую (системы счисления отличны от двоичной и десятичной), в четвертом – выполнить действия с числами в двоичной системе счисления, в пятом – составить таблицу истинности для заданного выражения. Каждое задание оценивается в один балл, за своевременно выполненную работу прибавляется еще один поощрительный балл, то есть максимальная оценка за своевременно выполненную контрольную работу составляет 6 баллов.

**Задание 1.** Переведите число из десятичной системы счисления в двоичную.

*Вариант 1:*  $125_{10} = x_2.$

*Вариант 2:*  $135_{10} = x_2.$

*Вариант 3:*  $145_{10} = x_2.$

*Вариант 4:*  $165_{10} = x_2.$

*Вариант 5:*  $175_{10} = x_2.$

*Вариант 6:*  $65_{10} = x_2.$

*Вариант 7:*  $75_{10} = x_2.$

*Вариант 8:*  $185_{10} = x_2.$

*Вариант 9:*  $195_{10} = x_2.$

*Вариант 10:*  $96_{10} = x_2.$

*Вариант 11:*  $149_{10} = x_2.$

*Вариант 16:*  $137_{10} = x_2.$

*Вариант 17:*  $144_{10} = x_2.$

*Вариант 18:*  $119_{10} = x_2.$

*Вариант 19:*  $117_{10} = x_2.$

*Вариант 20:*  $116_{10} = x_2.$

*Вариант 21:*  $97_{10} = x_2.$

*Вариант 22:*  $174_{10} = x_2.$

*Вариант 23:*  $121_{10} = x_2.$

*Вариант 24:*  $136_{10} = x_2.$

*Вариант 25:*  $194_{10} = x_2.$

*Вариант 26:*  $129_{10} = x_2.$

Вариант 12:  $148_{10} = x_2$ .

Вариант 13:  $154_{10} = x_2$ .

Вариант 14:  $193_{10} = x_2$ .

Вариант 15:  $183_{10} = x_2$ .

Вариант 27:  $128_{10} = x_2$ .

Вариант 28:  $148_{10} = x_2$ .

Вариант 29:  $155_{10} = x_2$ .

Вариант 30:  $52_{10} = x_2$ .

**Задание 2.** Переведите число из двоичной системы счисления в десятичную.

Вариант1:  $10000001_2 = x_{10}$ .

Вариант2:  $10000011_2 = x_{10}$ .

Вариант3:  $10101111_2 = x_{10}$ .

Вариант4:  $10001101_2 = x_{10}$ .

Вариант5:  $11001111_2 = x_{10}$ .

Вариант6:  $10001001_2 = x_{10}$ .

Вариант7:  $10101101_2 = x_{10}$ .

Вариант8:  $10011001_2 = x_{10}$ .

Вариант9:  $10101011_2 = x_{10}$ .

Вариант10:  $10101010_2 = x_{10}$ .

Вариант11:  $10000111_2 = x_{10}$ .

Вариант12:  $11101111_2 = x_{10}$ .

Вариант13:  $11100111_2 = x_{10}$ .

Вариант14:  $11000001_2 = x_{10}$ .

Вариант15:  $10001110_2 = x_{10}$ .

Вариант16:  $10001100_2 = x_{10}$ .

Вариант17:  $10001000_2 = x_{10}$ .

Вариант18:  $10000000_2 = x_{10}$ .

Вариант19:  $10100000_2 = x_{10}$ .

Вариант20:  $11100000_2 = x_{10}$ .

Вариант21:  $10011000_2 = x_{10}$ .

Вариант22:  $10000110_2 = x_{10}$ .

Вариант23:  $10000100_2 = x_{10}$ .

Вариант24:  $10001011_2 = x_{10}$ .

Вариант25:  $10001100_2 = x_{10}$ .

Вариант26:  $10000101_2 = x_{10}$ .

Вариант27:  $10101101_2 = x_{10}$ .

Вариант28:  $11001110_2 = x_{10}$ .

Вариант29:  $11001010_2 = x_{10}$ .

Вариант30:  $11001000_2 = x_{10}$ .

**Задание 3.** Найдите  $x$ :

Вариант 1:  $11010010_2 = x_{16}$ .

Вариант 2:  $12BD_{16} = x_2$ .

Вариант 3:  $11101110_2 = x_{16}$ .

Вариант 4:  $12A_{16} = x_2$ .

Вариант 5:  $10000111_2 = x_{16}$ .

Вариант 6:  $1C8_{16} = x_2$ .

Вариант 7:  $10011111_2 = x_{16}$ .

Вариант 8:  $1FF4_{16} = x_2$ .

Вариант 9:  $1000111_2 = x_{16}$ .

Вариант 10:  $2DF0_{16} = x_2$ .

Вариант 11:  $121D_{16} = x_2$ .

Вариант 16:  $101010_2 = x_{16}$ .

Вариант 17:  $10111100_2 = x_{16}$ .

Вариант 18:  $1AD6_{16} = x_2$ .

Вариант 19:  $10010101_2 = x_{16}$ .

Вариант 20:  $120F_{16} = x_2$ .

Вариант 21:  $12B_{16} = x_2$ .

Вариант 22:  $1AFD_{16} = x_2$ .

Вариант 23:  $10100_2 = x_{16}$ .

Вариант 24:  $1111100_2 = x_{16}$ .

Вариант 25:  $12F_{16} = x_2$ .

Вариант 26:  $10001111_2 = x_{16}$ .

Вариант 12:  $1001001_2 = x_{16}$ .    Вариант 27:  $1BC3_{16} = x_2$ .  
 Вариант 13:  $10FD_{16} = x_2$ .    Вариант 28:  $1111100_2 = x_{16}$ .  
 Вариант 14:  $1011111_2 = x_{16}$ .    Вариант 29:  $100110_2 = x_{16}$ .  
 Вариант 15:  $1001001_2 = x_{16}$ .    Вариант 30:  $11B_{16} = x_2$ .

**Задание 4.** Выполните сложение, вычитание и умножение чисел в двоичной системе счисления:

Вариант1:  $10001_2$  и  $111_2$ .    Вариант16:  $10101_2$  и  $1011_2$ .  
 Вариант2:  $10011_2$  и  $111_2$ .    Вариант17:  $10101_2$  и  $1000_2$ .  
 Вариант3:  $10111_2$  и  $111_2$ .    Вариант18:  $1101_2$  и  $11000_2$ .  
 Вариант4:  $11111_2$  и  $111_2$ .    Вариант19:  $10111_2$  и  $1011_2$ .  
 Вариант5:  $10001_2$  и  $101_2$ .    Вариант20:  $101111_2$  и  $1011_2$ .  
 Вариант6:  $10001_2$  и  $100_2$ .    Вариант21:  $111001_2$  и  $1001_2$ .  
 Вариант7:  $10001_2$  и  $11_2$ .    Вариант22:  $100001_2$  и  $1001_2$ .  
 Вариант8:  $11001_2$  и  $111_2$ .    Вариант23:  $11001_2$  и  $10011_2$ .  
 Вариант9:  $11001_2$  и  $1111_2$ .    Вариант24:  $111111_2$  и  $1101_2$ .  
 Вариант10:  $10001_2$  и  $1110_2$ .    Вариант25:  $10001_2$  и  $11011_2$ .  
 Вариант11:  $10111_2$  и  $111_2$ .    Вариант 26:  $111101_2$  и  $1101_2$ .  
 Вариант12:  $10001_2$  и  $101_2$ .    Вариант 27:  $11011_2$  и  $11001_2$ .  
 Вариант13:  $10011_2$  и  $1110_2$ .    Вариант 28:  $111101_2$  и  $1011_2$ .  
 Вариант14:  $1001_2$  и  $11100_2$ .    Вариант 29:  $1011_2$  и  $110001_2$ .  
 Вариант15:  $10101_2$  и  $1110_2$ .    Вариант30:  $111001_2$  и  $10001_2$

**Задание 5.** Составьте таблицу истинности для выражения:

Вариант 1:  $C = (A * \overline{B} + A) * \overline{B}$ ;  
 Вариант 2:  $C = \overline{A} * B + A * \overline{B * A}$ ;  
 Вариант 3:  $C = B * \overline{A} + A * \overline{B}$ ;  
 Вариант 4:  $C = \overline{A * B} + B + \overline{A}$ ;  
 Вариант 5:  $C = (A + B) * \overline{(A + B)}$ ;  
 Вариант 6:  $C = A * \overline{B} + B + A$ ;  
 Вариант 7:  $C = B * A + A * \overline{A} + A$ ;  
 Вариант 8:  $C = A * B + A * \overline{B * A}$ ;

- Вариант 9:  $C = (A * \overline{B + A + B}) * B$  ;
- Вариант 10:  $C = (\overline{A + B}) * (\overline{A + B})$  ;
- Вариант 11:  $C = \overline{B} * (B + \overline{A}) * \overline{A}$  ;
- Вариант 12:  $C = \overline{A + B + A * B}$  ;
- Вариант 13:  $C = \overline{B + A * A + B + \overline{A}}$  ;
- Вариант 14:  $C = \overline{A * B + \overline{A} + A + \overline{B}}$  ;
- Вариант 15:  $C = \overline{B * A * B + A}$  ;
- Вариант 16:  $C = \overline{A + B + A * B}$  ;
- Вариант 17:  $C = \overline{B} * A + B * \overline{A + B}$  ;
- Вариант 18:  $C = \overline{A * A + B + B * \overline{A + B}}$  ;
- Вариант 19:  $C = \overline{\overline{A + B} * B + A * \overline{B}}$  ;
- Вариант 20:  $C = \overline{A + B * B * A}$  ;
- Вариант 21:  $C = \overline{B + B * A * B}$  ;
- Вариант 22:  $C = \overline{A + B * A * A}$  ;
- Вариант 23:  $C = \overline{A + B * B * A}$  ;
- Вариант 24:  $C = \overline{A + B * A + B}$  ;
- Вариант 25:  $C = \overline{\overline{A + B} * A + B}$  ;
- Вариант 26:  $C = \overline{A + B * B + A * B}$  ;
- Вариант 27:  $C = \overline{A + B * B + A * B}$  ;
- Вариант 28:  $C = \overline{B * (B + A) * \overline{A}}$  ;
- Вариант 29:  $C = \overline{\overline{B} * (B + A) * A}$  ;
- Вариант 30:  $C = \overline{B * (B + A) * \overline{A}}$  .