

Глава 2. ПРОГРАММНОЕ УПРАВЛЕНИЕ КОМПЬЮТЕРОМ

- **Программа и данные**
- **Поколения и классификация ЭВМ**
- **Устройство компьютера и особенности обмена информацией между его узлами**
- **Архитектура фон Неймана**
- **Машинный код, выполнение машинной команды**
- **Типы программ**
- **Системы счисления. Перевод чисел из одной системы счисления в другую**
- **Представление чисел в памяти компьютера**
- **Машинная математика**

ПРОГРАММА И ДАННЫЕ

Программирование – это процесс создания программ.

Программа – набор инструкций, посланный вычислительной машине (компьютеру).

Носителем информации является *сообщение*.

Данные – это сообщения, закодированные в форму, пригодную для хранения и обработки их компьютером (на основе двоичного набора знаков – из-за простоты распознавания и хранения на физическом уровне сигналов, имеющих только два состояния: "включен – выключен").

Сейчас реализуется с помощью *CMOS-транзисторов*.

Для кодировки *сообщений* применяется двоичный набор, состоящий из двух знаков 0 и 1 (*binary digit*, сокращенно *bit*).

Порядок выполнения операций над данными строится на основе некоторого *алгоритма*.

Программу можно рассматривать как *алгоритм*, записанный на понятном для компьютера языке, и *данные*, которые компьютер будет обрабатывать в соответствии с этим алгоритмом.

ПОКОЛЕНИЯ ЭВМ

- **Первое** (электронные лампы – 1945-1955 гг.). Быстродействие – от неск. сотен до неск. тысяч операций в сек. Примеры: ENIAC, EDVAC, IBM 701; СССР – МЭСМ, БЭСМ.
- **Второе** (транзисторы – 1955-1965 гг.). Быстродействие – до сотен тыс. и миллионов операций в сек. Системное ПО, программирование на языках высокого уровня. Примеры: IBM 1620, PDP-1.
- **Третье** (интегральные схемы (ИС) – 1965-1980 гг.). Быстродействие – до дес. миллионов операций в сек. Массовое производство ИС. Примеры: IBM серии System/360, PDP-8; СССР – ЭВМ серии ЕС.
- **Четвертое** (сверхбольшие ИС – с 1980 г.). Микроэлектроника. Микропроцессор. Компьютерные сети. Примеры: ПК IBM PC, Apple Macintosh, суперкомпьютеры.
- **Пятое** (?). В настоящее время процесс продолжается в рамках парадигмы сверхбольшой ИС. Цель – "интеллектуализация" компьютера.

Переход к более высокому поколению – это принципиальное снижение размеров, потребляемой мощности, стоимости и увеличение быстродействия.

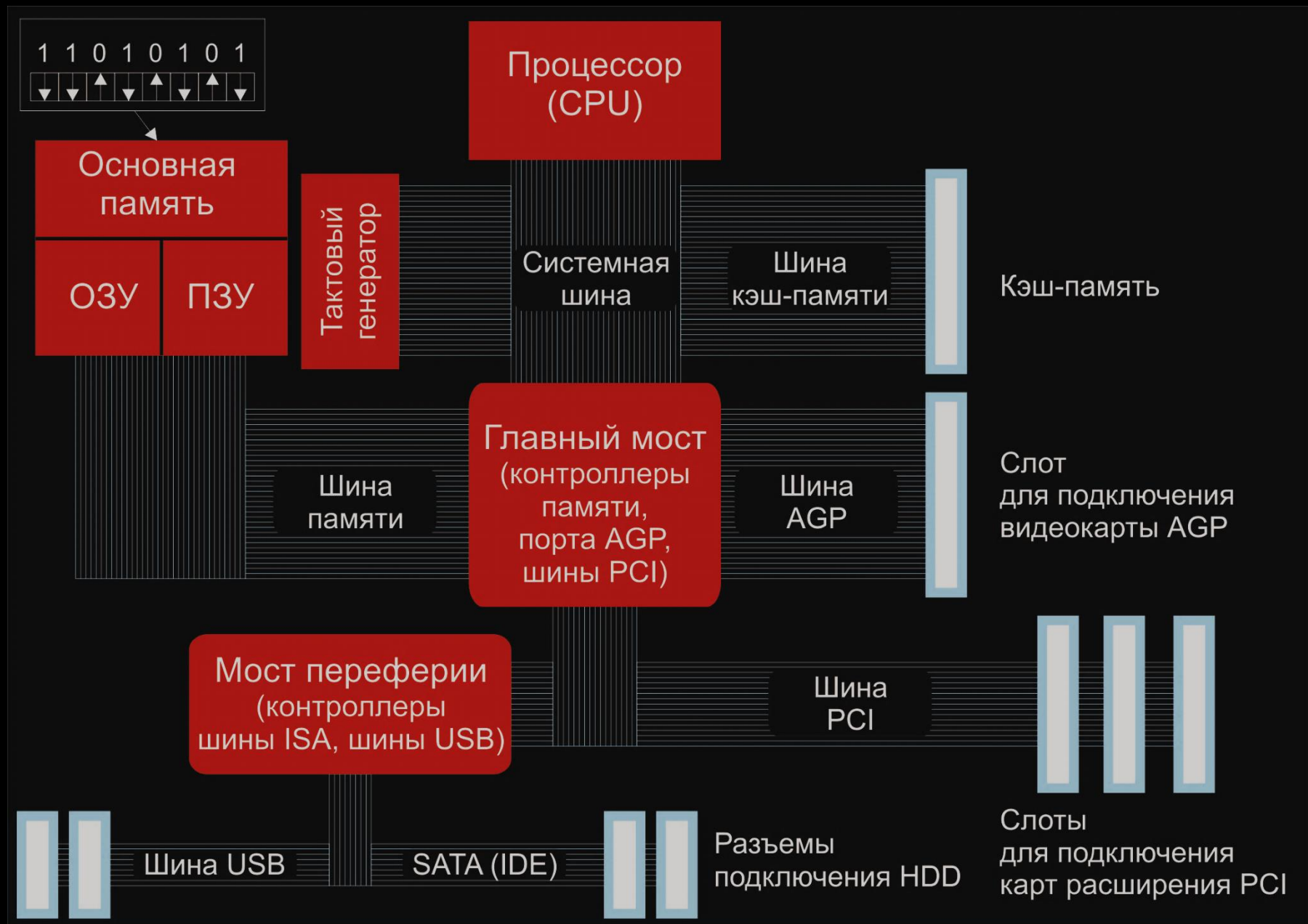
КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ



Мфлопс (MFLOPS, Million of Floating point Operation Per Second)

МИПС (MIPS, Million Instruction Per Second)

АРХИТЕКТУРА КОМПЬЮТЕРА



АРХИТЕКТУРА КОМПЬЮТЕРА (ОСНОВНЫЕ УЗЛЫ)

Процессор. Характеристики – *разрядность, тактовая частота, набор команд.*

Основная память (ОЗУ + ПЗУ). Объем памяти измеряется в *байтах*. **Байт** – группа из восьми бит, обрабатываемая как единое целое. Емкость ячейки памяти – 1 байт. Производные: Кбайт ($1024 (2^{10})$ байт), Мбайт ($1\ 048\ 576 (2^{20})$ байт) , Гбайт ($1\ 073\ 741\ 824 (2^{30})$ байт).

Чипсет (Chipset). В состав входят *главный мост и мост периферии.*

Компьютерные шины – набор линий-проводников для передачи информации между узлами. Выделяют *системную шину, периферийные шины*. Архитектура любой из шин включает линии для обмена данными (*шина данных*), для адресации данных (*шина адресов*), для управления данными (*шина управления*).

Пропускная способность шины (Мбайт/с) определяется ее *разрядностью*, умноженной на *тактовую частоту*.

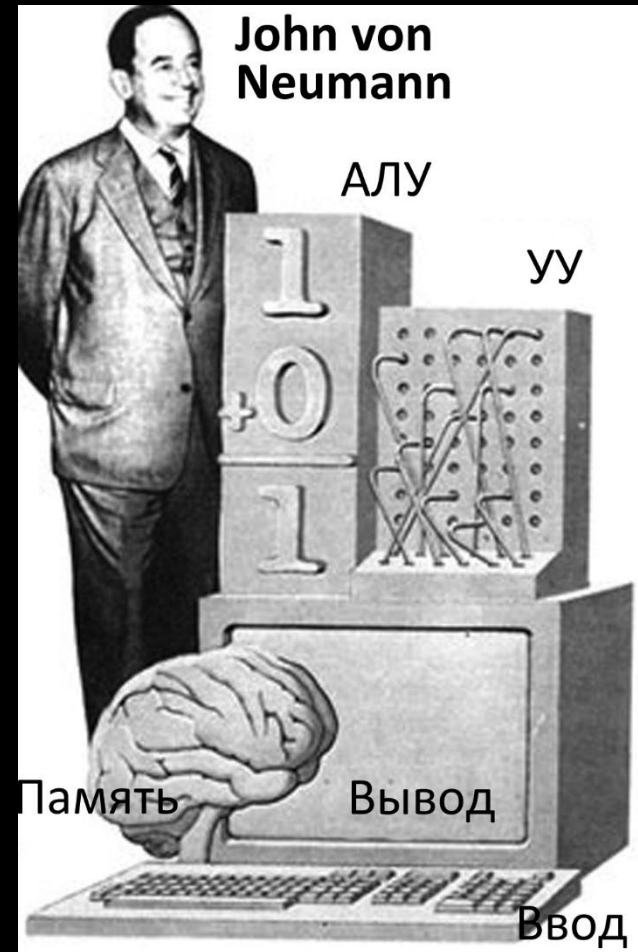
Кэш-память. Предназначена для временного хранения копий блоков данных тех областей ОЗУ, к которым выполнялись последние обращения и весьма вероятны обращения в ближайшие такты. Имеется кэш-память *1-го, 2-го, 3-го* уровней.

Другие узлы: магнитные диски, флэш-память, клавиатура, мышь, монитор, и т.д.

АРХИТЕКТУРА ФОН НЕЙМАНА

Машина (архитектура) фон Неймана построена на следующих принципах:

- **наличие трех основных элементов** – *памяти* (для хранения информации), *арифметико-логического устройства, АЛУ* (выполняет команды) и *устройства управления, УУ* (указывает команды для выполнения);
- **однородности памяти** – команды и данные хранятся в одной и той же памяти; вид записи команд и данных одинаков;
- **адресности** – структурно память состоит из пронумерованных ячеек, процессору в произвольный момент времени доступна любая ячейка;
- **программного управления** – программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности;
- **наличие канала связи** между памятью и процессором.



МАШИННЫЙ КОД

Машинная команда состоит из кода выполняемой операции (**оператор**) и адресной части (**операнды**).

КОД	АДРЕСНАЯ ЧАСТЬ
-----	----------------

Машинный код – закодированное представление команды процессора
7C90104A 00648B0D

Регистры – поименованные ячейки памяти процессора (сумматор – регистр АЛУ).

add	x
-----	---

одноадресная команда – содержимое ячейки x ОЗУ сложить с содержимым сумматора, а результат оставить в сумматоре;

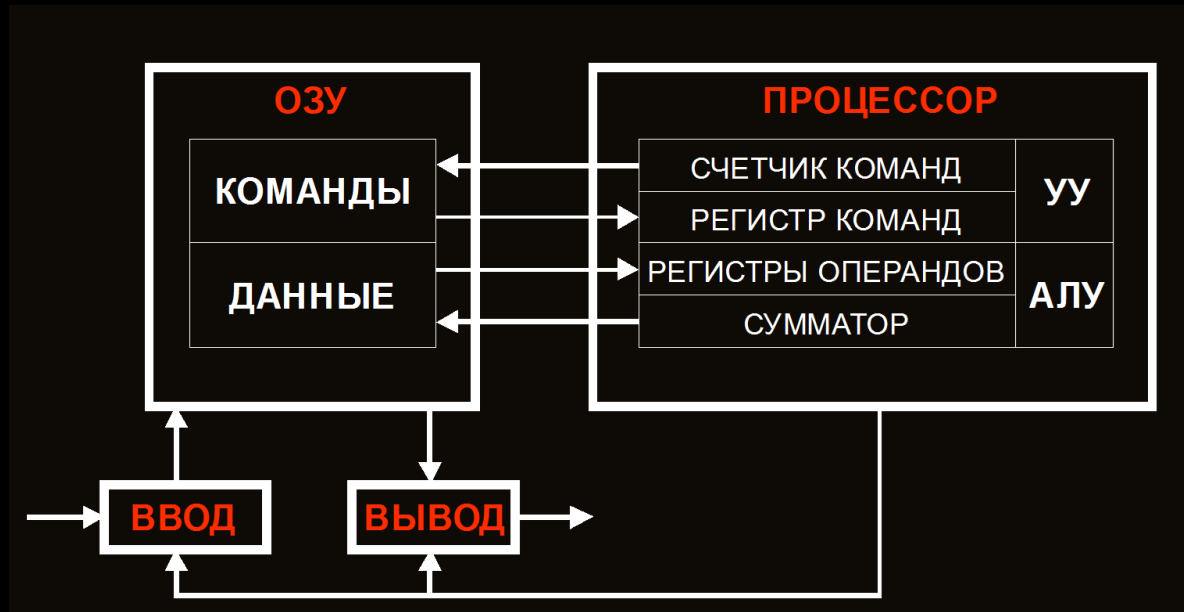
add	x	y
-----	---	---

двухадресная команда – сложить содержимое ячеек x и y, а результат поместить в ячейку y;

add	x	y	z
-----	---	---	---

трехадресная команда – содержимое ячейки x сложить с содержимым ячейки y, сумму поместить в ячейку z);

ВЫПОЛНЕНИЕ МАШИНОЙ КОМАНДЫ



1. из ячейки памяти, адрес которой хранится в счетчике команд, выбирается очередная команда; содержимое счетчика увеличивается на длину команды;
2. эта команда передается в УУ на регистр команд;
3. УУ расшифровывает адресное поле команды;
4. по сигналам УУ операнды считываются из памяти и записываются в АЛУ на регистры операндов;
5. УУ расшифровывает код операции и выдает в АЛУ сигнал выполнить соответствующую операцию над данными;
6. результат операции либо остается в процессоре, либо отправляется в память, если в команде был указан адрес результата.

ТИПЫ ПРОГРАММ



СИСТЕМЫ СЧИСЛЕНИЯ

Система счисления – способ отображения чисел с помощью символов.

Совокупность символов – алфавит, число символов в алфавите – размерность.

В позиционной системе счисления вес символа (цифры) зависит от его позиции

$V = B^N$, B – основание системы счисления, N – порядковый номер позиции.

Двоичная система счисления – позиционная (основание – 2).

Перевод числа из двоичной системы в десятичную:

$$(1101.1011)_2 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} + 1*2^{-4} = 13 + 0.5 + 0.125 + 0.0625 = (13.6875)_{10}$$

Перевод *целого числа* из десятичной системы в двоичную осуществляется последовательным делением на 2. В качестве остатка от деления получается очередная цифра двоичного числа, начиная с младшей:

$$13/2 = 6 \text{ (остаток 1 – младшая цифра)}, 6/2 = 3 \text{ (0)}, \\ 3/2 = 1 \text{ (1)}, 1/2 = 0 \text{ (1)}. \text{ Результат – } (1101)_2$$

Перевод *дроби* из десятичной системы в двоичную осуществляется умножением на 2. Целая часть полученного числа – очередная цифра двоичного, начиная с первой цифры после запятой:

$$0.6875*2 = 1.375 \text{ (первая цифра – 1)}, 0.375*2 = 0.75 \text{ (0)}, \\ 0.75*2 = 1.5 \text{ (1)}, 0.5*2 = 1.0 \text{ (1)}. \text{ Результат – } (0.1011)_2$$

СИСТЕМЫ СЧИСЛЕНИЯ

Восьмеричная система счисления – позиционная (основание – 8, используются цифры 0 1 2 3 4 5 6 7).

Шестнадцатеричная система счисления – позиционная (основание – 16, используются цифры 0 1 2 3 4 5 6 7 8 9 и первые буквы латинского алфавита A B C D E F).

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему (и обратно) осуществляется заменой каждой цифры эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

$$(537.1)_8 = 101\ 011\ 111.001 = (101\ 011\ 111.001)_2$$

5 3 7. 1

$$(1A3.F)_{16} = 0001\ 1010\ 0011.1111 = (1\ 1010\ 0011.1111)_2$$

1 A 3. F

"10"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
"2"	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
"8"	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
"16"	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ (ЦЕЛЫХ) В ПАМЯТИ

Представление (вид) числа в памяти определяется тем **типом**, к которому оно принадлежит. В частности, тип числа задает количество двоичных разрядов, отводимых под хранение числа.

Для хранения числа в памяти отводится целое число ячеек емкостью 1 байт – 8 бит (разрядов). Число занимает всю ячейку, независимо от того, сколько двоичных разрядов требуется для его представления.

Целые беззнаковые типы (представление числа 5)

BYTE (1 байт, 0...255)

0000 0101

WORD (2 байта, 0...65535)

0000 0000 0000 0101

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ (ЦЕЛЫХ) В ПАМЯТИ

Для хранения чисел со знаком старший разряд отводится под знак (0 – если число положительное, 1 – если отрицательное).

Целые знаковые типы (представление числа 5)

SHORTINT (1 байт, -128...127)

0 000 0101

SMALLINT (2 байта, -32768...32767)

0 000 0000 0000 0101

Отрицательные числа представлены в виде **дополнительного кода**.

Для получения дополнительного кода необходимо сначала поменять все разряды числа на обратные, а затем к полученному результату прибавить 1.

Целые знаковые типы (представление числа -5)

SHORTINT (1 байт, -128...127)

1 111 1011

SMALLINT (2 байта, -32768...32767)

1 111 1111 1111 1011

Все математические операции с числами основаны на сложении.

Сложение осуществляется поразрядно с соблюдением правила: "Если в результате сложения двух соответствующих разрядов чисел получилось число большее или равное основанию системы счисления (2), то следует из полученного числа отнять основание системы счисления и записать в строке итога полученный результат. Кроме того, необходимо запомнить единицу с тем, чтобы добавить ее при сложении следующего разряда".

$$\begin{array}{rcccccccc}
 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & & (73)_{10} \\
 + & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & & (37)_{10} \\
 \hline
 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & & (110)_{10}
 \end{array}$$

Вычитание заменяется сложением чисел, одно из которых берется с обратным знаком (используется дополнительный код).

$$\begin{array}{rcccccccc}
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & & (7)_{10} \\
 + & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & (-5)_{10} \\
 \hline
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & (2)_{10}
 \end{array}$$

Умножение заменяется сложением чисел, сдвинутых на разное число двоичных разрядов подобно тому, как это делается при умножении чисел "в столбик".

$$\begin{array}{r}
 \begin{array}{r}
 10.1 \\
 \times 1.1 \\
 \hline
 101 \\
 + 10.1 \\
 \hline
 11.11
 \end{array}
 \end{array}
 \begin{array}{l}
 (2.5)_{10} \\
 (1.5)_{10} \\
 \\
 (3.75)_{10}
 \end{array}$$

Деление выполняется через вычитание (путем многократного прибавления к делимому дополнительного кода делителя).

$$\begin{array}{r}
 1111 \mid 11 \\
 - 11 \\
 \hline
 11 \\
 - 11 \\
 \hline
 0
 \end{array}
 \qquad
 15 / 3 = 5$$

Возведение в степень выполняется через умножение и т.д.

Имеется **математический сопроцессор** – специальное устройство, ускоряющее вычисления с плавающей точкой.

МАШИННАЯ МАТЕМАТИКА

Работа компьютера (в том числе выполнение арифметических операций) основана на **логических действиях**.

Например, возможны только четыре комбинации соответствующих разрядов при сложении двух чисел, представленных в двоичном виде.

Правила по установке соответствующего разряда итогового числа построено по законам логики:

Разряд I слагаемого	Разряд II слагаемого
0	0
1	0
0	1
1	1

"Если складываются разряды с равным состоянием (ноль с нулем или единица с единицей), то итоговый разряд устанавливается равным нулю. В противном случае, он устанавливается равным единице. Если складываются два разряда, равные единице, то вырабатывается сигнал переноса единицы в следующий разряд".

Логическая конструкция "Если условие обращается в истину, то выполнить некую последовательность действий" называется **импликацией**.

Более сложную конструкцию: "Если ..., то..., иначе" реализуют с помощью оператора: **if – then – else**.