

Министерство образования Российской Федерации
Томский государственный университет
Факультет инновационных технологий

М.Н.Головчинер

ВВЕДЕНИЕ В АРХИТЕКТУРУ ЭВМ

Курс лекций

Томск - 2013

Оглавление

1. ВВЕДЕНИЕ	4
1.1. История развития ЭВМ.....	4
1.2. Принципы фон-Неймана	5
2. СИСТЕМЫ СЧИСЛЕНИЯ.....	5
2.1. Виды систем счисления. Позиционные системы	5
2.2. Двоичная арифметика	7
2.3. Восьмеричная и шестнадцатеричная системы счисления.....	9
2.4. Перевод чисел из одной системы счисления в другую.....	10
2.5. Формы представления чисел в ЭВМ	12
2.6. Сложение и вычитание в ЭВМ.....	13
3. БАЗОВЫЕ ФУНКЦИОНАЛЬНЫЕ ЭЛЕМЕНТЫ ЭВМ.....	17
3.1. Вентили, последовательностные логические схемы.....	17
3.2. Комбинационные схемы, понятие об интегральной схеме	21
4. ОБЩАЯ ОРГАНИЗАЦИЯ ЭВМ.....	26
4.1. Понятие архитектуры ЭВМ	26
4.2. Обобщенная структурно-функциональная схема ЭВМ.....	27
5. ПРОЦЕССОР	29
5.1. Команда ЭВМ	29
5.2. Общая структура процессора	30
6. ПОДСИСТЕМА ПАМЯТИ.....	34
6.1. Структура и характеристики памяти	34
6.2. Классы запоминающих устройств	35
6.2.1. Оперативное запоминающее устройство.....	36
6.2.2. Постоянное запоминающее устройство.....	37
6.2.3. Стековая память	37
6.2.4. Ассоциативные запоминающие устройства.....	41
6.2.5. Кэш-память	42
7. ПОДСИСТЕМА УПРАВЛЕНИЯ.....	47
7.1. Основные функции и состав устройства управления	47
7.2. Принципы функционирования	49
7.3. Понятие микропрограммирования.....	51
8. УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ	52
8.1. Системная магистраль "Общая шина". Контроллеры внешних устройств и уровни сопряжения	52
8.2. Управление магистралью "Общая шина"	55
8.3. Программно-управляемая передача информации	57
9. СИСТЕМА ПРЕРЫВАНИЙ.....	58
9.1. Основные понятия. Типы прерываний	58

9.2	Общая организация прерываний.....	59
9.3	Организация системы прерываний с использованием векторов прерываний	60
9.4	Организация прямого доступа к памяти	61
10.	ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ.....	63
10.1.	Схема взаимодействия устройств в ЭВМ. Операции обмена информацией.....	63
10.2.	Назначение и классификация периферийных (внешних) устройств.....	64
10.3.	Внешние запоминающие устройства большой емкости.....	65
10.3.1.	Накопители с жесткими магнитными дисками.....	65
10.3.2.	Оптические диски	71
10.3.3.	Flash-память.....	72
10.3.4.	Стримеры	74
10.4.	Средства ввода-вывода информации.....	74
10.4.1.	Устройства ввода	76
10.4.2.	Устройства вывода.....	84
10.4.3.	Устройства обмена данными (средства телекоммуникации).....	105
11.	Устройство персонального компьютера	105
11.1.	Состав компьютера.....	106
11.2.	Контроллеры и адаптеры. Общий состав устройств.....	106

1. ВВЕДЕНИЕ

1.1. История развития ЭВМ

Этапными в появлении и развитии вычислительной техники можно считать следующие даты:

- 1623 г. Вильгельм Шиккард создает первую из известных в настоящее время счетных машин, включающую механизмы выполнения операций "+" и "-".
- 1641 г. Выдающийся математик Блез Паскаль создает счетную машину, автоматизирующую выполнение операций "+" и "-". Идеи Паскаля оказали очень мощное влияние на дальнейшее развитие вычислительной техники.
- 1673 г. Великий Готфрид Лейбниц создает счетную машину (калькулятор, прототип арифмометра), реализующую операции "+", "-", "x", ":".
- 1821 г. Выпуск Карлом Томасом на основании идей Лейбница устройства, названного им *арифмометром*. С этого времени началось практическое применение вычислительных устройств.

На протяжении XIX в. было создано много конструкций счетных машин, повысилась точность и надежность их работы, удобство общения. Возникли идеи фиксации результатов счета путем выдачи их на печать, появилась клавиатура, кнопки гашения результатов и т.п.

Принципиально же новые концепции связаны с именем крупного английского математика Чарльза Бэбиджа. Эти концепции предвосхитили основные идеи в создании современных вычислительных машин.

"Аналитическая" машина Бэбиджа должна была состоять из следующих основных блоков:

- Устройство для хранения чисел на регистрах из зубчатых колес, включающее систему передачи этих чисел от одного узла машины к другому ("склад").
- Устройство для осуществления арифметических операций над числами, взятыми со "склада" ("фабрика").
- Устройство для управления последовательностью действий машины ("контора").
- Устройства для ввода исходных данных и вывода результатов на печать.

Все устройства в машине Бэбиджа были задуманы как чисто механическими, с использованием перфокарт, которые изобрел и в 1804 г. впервые применил Жозеф Мари Жаккар для управления ткацким станком.

Следует отметить, что отцом счетно-перфорационной техники, непосредственно предшествующей современным машинам, является Герман Холлерит. В 1896 г. Холлерит основал фирму по выпуску перфокарт и счетно-перфорационной техники - **IBM** (International Business Machine).

Перфокарты предназначались в машине Бэбиджа для задания типа операций, выборки из "склада" чисел, над которыми необходимо произвести действие, в "фабрику" и, соответственно, обратного переноса результата выполнения операции. Посредством перфокарт Бэбидж предполагал управлять и специальными устройствами ввода данных и вывода результатов.

Карты программы (последовательности перфокарт с указанными операциями), в зависимости от знака результата на арифметическом устройстве, могли двигаться вперед и назад, и тем самым открывалась возможность *менять программу в зависимости от результатов вычислений*. Это - главное отличительное свойство универсальных вычислительных машин от других вычислительных устройств. Именно это гениальное открытие Бэбиджа дало возможность использовать вычислительные машины как инструмент аналитических исследований. Таким образом, Бэбидж предвосхитил идею и принципы организации программно-управляемой автоматической машины, предназначенной для выполнения различных вычислений.

Работа Бэбиджа была доведена до инженерного проекта, однако технический уровень не позволил осуществить ее реализацию.

С аналитической машиной Бэбиджа связано и зарождение программирования. Прекрасный математик, дочь великого поэта лорда Байрона леди Ада Лавлейс исследовала возможность использования аналитической машины для выполнения сложных вычислений, то есть в современной терминологии проанализировала и разработала первые программы, доказав универсальные возможности вычислительных машин.

Замечание. В честь леди Лавлейс один из известных языков программирования назван «Ада».

Труды Бэбиджа были опубликованы после его смерти в 1888 г. и забыты.

А первые универсальные цифровые вычислительные машины (ЦВМ), в основу разработки которых легли идеи Ч.Бэбиджа, были созданы в 30-40-х годах XX столетия.

1.2. Принципы фон-Неймана

Анализ сильных и слабых сторон первых ЦВМ позволил коллективу исследователей, в состав которого входил выдающийся математик Джон фон-Нейман, сформулировать основные концепции организации электронных ЦВМ. Поскольку впервые публично официально эти идеи озвучил лично фон-Нейман, ЦВМ, созданные на основе этих принципов, часто называют *машинами фон-Неймана*. Перечень основных принципов фон-Неймана включает:

1. Необходимость использования двоичной системы счисления для представления информации в ЦВМ.
2. Наличие единого запоминающего устройства ("склада") с последовательно нумеруемыми ячейками.
3. Отсутствие явного различия между командами и данными (принцип "хранимой программы").
4. Строго последовательное выполнение команд (операций) программы.
5. Назначение данных (их смысл, семантика) не является их составной частью, а определяется только логикой программы, их обрабатывающей.

Заметим, что перечисленные принципы являются основными и далеко не исчерпывают всех предложений фон-Неймана.

Разработка ЦВМ на основе элементов с двумя устойчивыми состояниями дала возможность широкого применения для построения электронных схем аппарата двоичной (бинарной) логики, разработанной в середине XIX в. английским математиком Джорджем Булем.

Алгебра логики нашла свое применение и в области программирования (например, разработка алгоритмов).

Принцип "хранимой программы" позволяет оперировать с закодированной в двоичном виде программой как с числами, что дает возможность ее модифицировать во время выполнения.

Все эти концепции и совершенствование технологии привели к созданию в конце 40-х - начале 50-х годов XX в. первого поколения серийных электронно-вычислительных машин (ЭВМ).

2. СИСТЕМЫ СЧИСЛЕНИЯ

2.1. Виды систем счисления. Позиционные системы

Системой счисления называется совокупность приемов и правил для наименования и обозначения чисел. Условные знаки, применяемые для обозначения чисел, называются *цифрами*. Будем предполагать, что количество цифр, то есть алфавит, на основе которого составляются числа в некоторой системе счисления, конечно.

Пусть S - некоторое целое число, большее единицы, которое будем называть *основанием системы счисления*. Выберем S попарно различных знаков - S -ичных цифр. Выберем, кроме того, S последовательных целых чисел, среди которых содержится нуль. Эту последовательность чисел будем называть *базой* системы счисления. Между S -ичными цифрами и числами базы установим взаимно-однозначное соответствие.

Обычно все системы счисления разбивают на два класса: **непозиционные** и **позиционные**.

Непозиционной называют систему счисления, в которой значение каждой цифры в любом месте последовательности цифр, означающей запись числа, не изменяется. Примером непозиционной системы счисления служит римская нумерация.

Системы, в которых значение каждой цифры зависит и от места в последовательности цифр при записи числа, носят название *позиционных*. Позиционной является самая распространенная десятичная система счисления. Далее в тексте рассматриваются только позиционные системы счисления.

В позиционной системе счисления число записывают в виде последовательности S -ичных цифр, которая разделена запятой на две последовательности.

Запись числа может иметь вид:

$$A_n A_{n-1} \dots A_1 A_0, A_{-1} \dots A_{-m}, (*)$$

где A_i ($i=n, \dots, -m$) – S -ичные цифры.

Считается, что позиции, на которых в последовательности (*) стоят цифры, перенумерованы:

- позиции, расположенные левее запятой, перенумерованы подряд справа налево числами нуль, один, два и т.д.;
- позиции, расположенные справа от запятой, перенумерованы подряд слева направо с помощью чисел минус один, минус два и т.д.

Эти перенумерованные позиции называются S -ичными разрядами числа.

Каждой цифре последовательности (*) приписано определенное *значение*. Цифра, стоящая в нулевом разряде, имеет своим значением соответствующее ей число базы. Цифра, стоящая в некотором разряде, имеет значение, в S раз большее того, которое она имела бы в разряде с номером, меньшим на единицу (или значение, в S раз меньшее того, которое она имела бы в разряде с номером, большим на единицу).

Последовательность S -ичных цифр обозначает число, равное сумме значений ее цифр, таким образом, в соответствии со сказанным, последовательность S -ичных цифр (*) обозначает число

$$A_n S^n + A_{n-1} S^{n-1} + \dots + A_1 S^1 + A_0 S^0 + A_{-1} S^{-1} + \dots + A_{-m} S^{-m} (**)$$

В дальнейшем вместо слов "запись числа" будем говорить просто "число".

Последовательность цифр числа, расположенных слева от запятой, представляет *целую часть* числа. Последовательность, которая получится, если целую часть числа заменить нулем, называется его *дробной частью*.

Сложение, вычитание, умножение и деление чисел, представленных в S -ичной позиционной системе, выполняются с использованием таблиц сложения, вычитания и умножения, подобно тому, как это делают в десятичной системе счисления.

Отметим, что умножение числа на основание системы S , как это следует из формулы (**), сводится к переносу запятой на один разряд вправо, а деление на S - к переносу запятой на один разряд влево.

Ниже рассматриваются важнейшие, наряду с десятичной, позиционные системы счисления, используемые при подготовке и решении задач на ЭВМ.

2.2. Двоичная арифметика

Будем рассматривать двоичную систему счисления с цифрами $\{0,1\}$. Именно эта система счисления получила широкое применение в вычислительных машинах. Ниже приводятся правила выполнения арифметических операций с двоичными числами.

Сложение. Таблица сложения имеет вид:

$$\begin{aligned}0 + 0 &= 0 \\0 + 1 &= 1 \\1 + 0 &= 1 \\1 + 1 &= 10\end{aligned}$$

Последняя сумма представляет собой двузначное число. Это следует понимать как перенос одной двоичной единицы в соседний старший разряд. Это можно записать так:

$$1 + 1 = 0 + \text{перенос единицы в соседний старший разряд.}$$

Пример. Сложить двоичные числа $(X)_2 = 110,1011$ и $(Y)_2 = 10111,10101$.

По правилам сложения столбиком имеем:

$$\begin{array}{r}+ 110,1011 \\10111,10101 \\ \hline^10001,00011 \quad \text{- поразрядная сумма без учета переносов} \\+ 1 \quad 1 \quad 1 \quad \text{- переносы} \\ \hline11100,01011 \quad \text{- поразрядная сумма без учета повторных переносов} \\+ \quad 1 \quad \text{- повторные переносы} \\ \hline11110,01011 \quad \text{- окончательная сумма}\end{array}$$

Вычитание. Таблица вычитания имеет вид

$$\begin{aligned}0 - 0 &= 0 \\1 - 0 &= 1 \\1 - 1 &= 0 \\10 - 1 &= 1\end{aligned}$$

При необходимости, когда в некотором разряде приходится вычитать единицу из нуля, заимается единица из ближайшего старшего разряда, где она имеется.

Пример. Вычесть из $(X)_2 = 11010,1011$ число $(Y)_2 = 1101,01111$

Имеем:

$$\begin{array}{r}- 11010,1011 \\1101,01111 \\ \hline1101,00111\end{array}$$

Умножение. Таблица умножения имеет очевидный вид

$$\begin{aligned}0 \times 0 &= 0 \\0 \times 1 &= 0 \\1 \times 0 &= 0 \\1 \times 1 &= 1\end{aligned}$$

Умножение двух двоичных чисел выполняется так же, как и умножение десятичных. Сначала получают частичные произведения и затем их суммируют с учетом веса соответствующего разряда множителя.

Умножение производится, начиная с младшего или старшего разряда множителя, что и определяет направление сдвига. Если сомножители имеют дробные части, то положение запятой в произведении определяется по тем же правилам, что и для десятичных чисел.

Пример. Перемножить двоичные числа $(X)_2 = 101,1101$ и $(Y)_2 = 1001,101$
Имеем:

$$\begin{array}{r}
 101,1101 \\
 \times 1001,101 \\
 \hline
 1011101 \\
 0000000 \\
 1011101 \\
 1011101 \\
 0000000 \\
 0000000 \\
 1011101 \\
 \hline
 110111111001
 \end{array}$$

Искомый результат 110111,111001

На примере наглядно иллюстрируется отличительная особенность умножения в двоичной системе счисления – это его простота, обусловленная простотой таблицы умножения. В соответствии с ней, каждое частичное произведение или равно нулю, если в соответствующем разряде множителя стоит нуль, или равно множимому, сдвинутому на соответствующее число разрядов, если в соответствующем разряде множителя стоит единица. Таким образом, операция умножения в двоичной системе сводится к операциям сдвига и сложения.

Очевидно, что тот же результат получится, если начать умножение со старших разрядов множителя.

Деление. Деление чисел в двоичной системе производится аналогично делению десятичных чисел. Деление начинается с того, что от делимого слева отделяется минимальная группа разрядов, которая, рассматриваемая как число, превышает или равна делителю. Дальнейшие действия выполняются по обычным правилам, причем последняя целая цифра частного получается тогда, когда все цифры делимого исчерпаны. Очевидно, что достаточно рассмотреть только деление целых чисел, поскольку делимое и делитель всегда могут быть приведены к целому виду путем перенесения запятой в делимом и делителе на одинаковое количество разрядов вправо и дописывания необходимого количества нулей.

Пример. Разделить $(X)_2 = 11011,11$ на число $(Y)_2 = 10,111$
Имеем:

$$\begin{array}{r}
 11011110 \quad | \quad 10111 \\
 - 10111 \quad \quad 1001 - \text{частное} \\
 \hline
 100110 \\
 - 10111 \\
 \hline
 1111 - \text{остаток}
 \end{array}$$

Таким образом, выполнение арифметических операций в двоичной системе счисления достаточно просто (особенно при выполнении операций сложения, вычитания и умножения). Благодаря этому, применение двоичной системы в вычислительных машинах позволяет упростить схемы устройств, в которых осуществляются арифметические операции над числами.

2.3. Восьмеричная и шестнадцатеричная системы счисления

Причины, по которым данные системы счисления широко используются в литературе (особенно, в документации) по вычислительной технике, будут обсуждены несколько позднее; здесь приведем лишь основные сведения об этих системах.

Восьмеричная система счисления. Алфавит системы образуют цифры {0, 1, 2, 3, 4, 5, 6, 7}.

Основание системы (восемь) записывают как 10. Ниже представлены таблицы сложения и умножения в восьмеричной системе, а так же примеры выполнения арифметических операций.

Таблица сложения

Таблица умножения

+	0	1	2	3	4	5	6	7	10		×	0	1	2	3	4	5	6	7	10	
0	0	1	2	3	4	5	6	7	10		0	0	0	0	0	0	0	0	0	0	0
1	1	2	3	4	5	6	7	10	11		1	0	1	2	3	4	5	6	7	10	
2	2	3	4	5	6	7	10	11	12		2	0	2	4	6	10	12	14	16	20	
3	3	4	5	6	7	10	11	12	13		3	0	3	6	11	14	17	22	25	30	
4	4	5	6	7	10	11	12	13	14		4	0	4	10	14	20	24	30	34	40	
5	5	6	7	10	И	12	13	14	15		5	0	5	12	17	24	31	36	43	50	
6	6	7	10	11	12	13	14	15	16		6	0	6	14	22	30	36	44	52	60	
7	7	10	11	12	13	14	15	16	17		7	0	7	16	25	34	43	52	61	70	
10	10	11	12	13	14	15	16	17	20		10	0	10	20	30	30	50	60	70	100	

Примеры выполнения арифметических операций.

а) сложение

$$\begin{array}{r} 327,71102 \\ + \quad 35,67735 \\ \hline 365,61037 \end{array}$$

б) вычитание

$$\begin{array}{r} 11076,01 \\ - \quad 705,62 \\ \hline 10170,17 \end{array}$$

в) умножение

$$\begin{array}{r} \times 173,263 \\ \quad 16,35 \\ \quad 1140577 \\ \quad 562031 \\ 1344062 \\ \hline 173263 \\ \hline 3366,52307 \end{array}$$

г) деление

$$\begin{array}{r} 3366,52307 \quad | \quad 16,35 \\ - 1635 \quad 173,263 \\ \hline 15315 \\ - 14513 \\ \hline 6022 \\ - 5327 \\ \hline 5033 \\ - 3472 \\ \hline 13410 \\ - 12656 \\ \hline 5327 \\ - 5327 \\ \hline 0000 \end{array}$$

Шестнадцатеричная система счисления. Алфавит системы включает шестнадцать символов

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.

Основание системы (шестнадцать) записывается как 10. Ввиду большого объема представление таблиц сложения и умножения здесь опускается; их нетрудно составить по образцу таблиц для восьмеричной системы.

2.4. Перевод чисел из одной системы счисления в другую

Простой случай. Наиболее прост перевод чисел из r -ичной позиционной системы в q -ичную, если имеет место соотношение $r=q^k$ (k - целое положительное число).

В этом случае перевод из r -ичной системы счисления в q -ичную производят "поразрядно", заменяя каждую r -ичную цифру равным ей k -разрядным числом, записанным в q -ичной системе счисления.

Перевод из q -ичной системы в r -ичную при этом производят следующим образом. Двигаясь от запятой вправо и влево, разбивают q -ичную запись числа на группы по k цифр. Если при этом самая левая или самая правая группы окажутся неполными, к ним приписывают соответственно слева и справа столько нулей, чтобы каждая из них содержала k цифр. После этого каждую группу q -ичных цифр заменяют одной r -ичной цифрой, равной числу, обозначенному этой группой q -ичных цифр.

Рассмотрим процесс перевода чисел между системами счисления с основаниями 2, 8 (2^3) и 16 (2^4). Алгоритм перевода основан на использовании следующей таблицы:

2-я система	8-я система	16-я система
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	10	8
1001	11	9
1010	12	A
1011	13	B
1100	14	C
1101	15	D
1110	16	E
1111	17	F

Перевод чисел из восьмеричной системы в двоичную. Каждая цифра восьмеричного числа заменяется соответствующей двоичной триадой (трехзначным числом).

Пример. $273,54_8 = (010) (111) (011), (101) (100)_2 = 10111011,1011_2$

Перевод чисел из двоичной системы в восьмеричную. Искомое двоичное число делится на двоичные триады согласно приведенным выше правилам, а затем каждая триада заменяется соответствующей восьмеричной цифрой.

Пример. $11011,0011_2 = 11 (011), (001) 1_2 = (011) (011), (001) (100)_2 = 33,43_8$

Перевод чисел из шестнадцатеричной системы в двоичную. Каждая цифра шестнадцатеричного числа заменяется соответствующей двоичной тетрадой (четырёхзначным двоичным числом).

Пример. $A_5, B_{16} = (1010)(0101), (1011)(0001)_2 = 10100101, 10110001_2$

Перевод чисел из двоичной системы в шестнадцатеричную. Искомое двоичное число делится на двоичные тетрады, а затем каждая тетрада заменяется соответствующей шестнадцатеричной цифрой.

Пример.

$$1001110, 011011_2 = 100(1110), (0110)_{16} = (0100)(1110), (0110)(1100)_2 = 4E, 6C_{16}$$

Перевод чисел из восьмеричной системы счисления в шестнадцатеричную и наоборот. Такой перевод проще всего осуществлять, используя в качестве промежуточного этапа перевод в двоичную систему счисления. То есть:

$$8 \rightarrow 16 \text{ можно заменить } 8 \rightarrow 2 \rightarrow 16$$

$$16 \rightarrow 8 \text{ можно заменить } 16 \rightarrow 2 \rightarrow 8$$

Общий случай. Перевод числа из r -ичной в q -ичную систему счисления, когда $r \neq q^k$, производится отдельно для целой и дробной частей числа. Теоретическое обоснование приводимых алгоритмов опустим.

Перевод целой части числа. Число записывается в r -ичной системе счисления. Производится деление в r -ичной системе целой части подвергающегося переводу числа на q ; в остатке получается число, равное последней цифре искомой q -ичной записи. Полученное частное снова делится на q , в остатке получается число, равное предпоследней цифре q -ичной записи, и т.д. Процесс повторяется до тех пор, пока в частном не будет получено число, меньшее, чем q , которое равно первой цифре q -ичной записи.

Пример. Перевод числа 191 из десятичной в восьмеричную и шестнадцатеричную системы счисления.

а) перевод в восьмеричную систему

$$\begin{array}{r} 191 \overline{) 8} \\ \underline{16} 23 \overline{) 8} \\ 31 \underline{16} 2 \\ \underline{24} 7 \\ 7 \end{array}$$

б) перевод в шестнадцатеричную систему:

$$\begin{array}{r} 191 \overline{) 16} \\ \underline{16} 11_{10} = B_{16} \\ \underline{31} \\ \underline{16} \\ 15_{10} = F_{16} \end{array}$$

Итоговый результат: $191_{10} = 277_8 = BF_{16}$

Перевод дробной части числа. Число q записывают в r -ичной системе счисления. Умножают в r -ичной системе дробь на q . Целая часть произведения равна первой цифре q -ичной записи дроби. Дробную часть произведения снова умножают на q . Целая часть произведения равна следующей цифре q -ичной записи дроби. Процесс продолжают либо до тех пор, пока он не закончится получением целого произведения, либо до тех пор, пока не получат требуемого количества цифр q -ичной записи дроби.

Пример. Перевод десятичного числа 0,6875 в восьмеричную и шестнадцатеричную системы.

а) перевод в восьмеричную систему:

$$\begin{array}{r} 0 \overline{) 6875} \\ \underline{8} \\ 5 \overline{) 5000} \\ \underline{8} \\ 4 \overline{) 0000} \end{array}$$

а) перевод в шестнадцатеричную систему:

$$\begin{array}{r} 0 \overline{) 6875} \\ \underline{16} \\ B \overline{) 0000} \end{array}$$

Итоговый результат: $0,6875_{10} = 0,54_8 = 0,B_{16}$

Комбинируя приведенные правила, можно без труда произвести перевод числа из любой позиционной системы счисления в любую другую позиционную систему счисления.

В силу того, что выполнение деления в системах счисления, отличных от десятичной, для нас непривычно, перевод из других систем счисления в десятичную часто производят с помощью формулы (**). В ее правой части р-ичные цифры, само число р и показатели степеней записывают в десятичной системе счисления, после чего вычисляют значение полученного выражения уже по правилам действий, принятых в десятичной системе.

Пример. Перевести число $277,54_8$ в десятичную систему. По формуле (**):

$$277,54_8 = (2 \cdot 10^2 + 7 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 4 \cdot 10^{-2})_8 =$$

$$(2 \cdot 8^2 + 7 \cdot 8^1 + 7 \cdot 8^0 + 5 \cdot 8^{-1} + 4 \cdot 8^{-2})_{10} = (2 \cdot 64 + 7 \cdot 8 + 7 + 5/8 + 4/64)_{10} = 191,6875_{10}$$

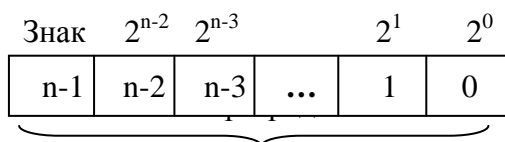
Итак, восьмеричное число $277,54$ равно десятичному числу $191,6875$.

2.5. Формы представления чисел в ЭВМ

Обычно числа в машинах представляются в одной из двух форм: с фиксированной запятой или с плавающей запятой.

Представление чисел с фиксированной запятой. В ячейке памяти ЭВМ число записывается последовательностью изображающих его двоичных цифр. Количество разрядов, отведенное для записи дробной и целой частей чисел, строго фиксировано. Это позволяет запятую в изображении числа опускать. Для изображения знака числа обычно отводится самый левый разряд ячейки, причем "плюс" кодируется двоичной цифрой 0, а "минус" - 1.

На практике при представлении чисел с фиксированной запятой запятая "закрепляется" после самого правого разряда числа, то есть все числа **целые**.



При данном представлении может быть записано число N такое, что $0 \leq |N| \leq 2^{n-1} - 1$.

Очевидно, что при выполнении арифметических операций над числами с фиксированной запятой может произойти переполнение разрядной сетки, то есть превышение значения результата верхней границы $- 2^{n-1} - 1$.

Представление чисел в форме с плавающей запятой. При форме представления с плавающей запятой число N считают приведенным к виду

$$N = m \cdot P^n, \quad (***)$$

где:

m - число с запятой, фиксированной после определенного разряда, называемое **мантиссой**,

n - целое число, называемое **порядком**,

P - основание системы счисления.

Запятую в мантиссах, как правило, фиксируют перед старшим (цифровым) разрядом, то есть в качестве мантисс берут правильные дроби. Это связано с удобством умножения мантисс (их произведение будет при этом также правильной дробью).

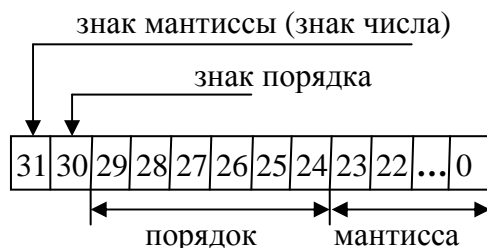
Если в записи (***) старшая цифра мантиссы отлична от нуля, то число N называют нормализованным (в противном случае - ненормализованным). Таким образом, нормализованными являются числа, для которых $1/P \leq |m| < 1$.

Пример. Десятичное число $25,2$ в форме с плавающей запятой (при запятой мантиссы, фиксированной перед старшим разрядом) может иметь любой из видов

$$0,252 \cdot 10^2 \quad 0,0252 \cdot 10^3 \quad 0,00252 \cdot 10^4$$

и т.д. Нормализованной является только запись $0,252 \cdot 10^2$, а все остальные являются ненормализованными.

Ниже изображено возможное представление нормализованных чисел с плавающей запятой в 32-х разрядной сетке ЭВМ.



При обработке в ЭВМ чисел с плавающей запятой возможно переполнение разрядной сетки, которое заключается в том, что в результате какой-либо операции возникает число, имеющее порядок с большей разрядностью, чем допустимая при представлении порядка в машине. Кроме того, в машине возможно появление машинных нулей, т.е. нормализованных чисел, отличных от нуля, но имеющих порядок, меньший самого малого порядка, представимого в разрядной сетке. Обычно машинные нули записываются в виде нулевой мантииссы и нулевого порядка.

Если количество разрядов мантииссы больше количества, выделенного для ее представления в разрядной сетке ЭВМ, избыточные младшие разряды отбрасываются, возможно, с округлением последнего из оставшихся.

Диапазон нормализованных двоичных чисел с плавающей запятой, представленных в разрядной сетке ЭВМ, если запятые в мантииссах фиксированы перед старшими разрядами, оценивается неравенствами

$$2^{-2s} \leq |N| \leq (1 - 2^{-k}) \cdot 2^{s-1},$$

где N - исходное число,

k - количество цифровых разрядов мантииссы,

s - количество цифровых разрядов порядка.

2.6. Сложение и вычитание в ЭВМ

Операции над числами с фиксированной запятой. Операция вычитания в машине реализуется достаточно сложно, так как при этом часто возникает необходимость "занимания" единиц в соседнем старшем разряде. Если же в соседнем старшем разряде этого сделать нельзя, необходим заем в следующем по старшинству разряде и т.д. Если же уменьшаемое меньше вычитаемого, этот процесс будет продолжаться до самого старшего разряда, а затем нужно вычитание делать заново, поменяв знак разности и вычитаемое и уменьшаемое местами.

Упрощение операции вычитания реализуется сведением ее к операции сложения, для чего применяют специальные способы представления отрицательных чисел.

Далее рассматриваются целые двоичные числа вида

$$N = \pm a_{n-2}a_{n-3}\dots a_0,$$

записанные в ячейку памяти ЭВМ, где

$$a_i = \{0, 1\}, i = 0, \dots, n-2 \text{ – цифры числа,}$$

n – разрядность (количество разрядов) ячейки памяти.

Прямой код. В ячейке число N представляется (кодируется) следующим образом:

$$[N]_{\text{пр}} = \begin{cases} 0 a_1 a_2 \dots a_n, & N \geq 0 \\ 1 a_1 a_2 \dots a_n, & N \leq 0 \end{cases}$$

Запись $[N]_{\text{пр}}$ называется **прямым кодом** числа N .

Пример. Записать в прямом коде двоичные числа $X = 101110$ и $Y = -101110$:

$$[X]_{\text{пр}} = 0101110$$

$$[Y]_{\text{пр}} = 1101110$$

Достоинство прямого кода состоит в том, что в нем достаточно просто реализуются алгоритмы выполнения «длинных» операций – умножения и деления.

Недостатками прямого кода являются:

1) двойное представление нуля:

$$00000000 = +0$$

$$10000000 = -0,$$

которое должно быть учтено либо аппаратными, либо программными средствами;

2) представление операнда в виде двух частей (знака и значащей компоненты) приводит к достаточно сложным алгоритмам операций сложения/вычитания в процессоре.

Дополнительный код

Дополнительный код $[N]_{\text{д}}$ целого положительного числа N совпадает с его прямым кодом

Дополнительный код целого отрицательного числа получается в результате следующей последовательности действий:

1. дополнение искомого числа до модуля (отсюда и название кода);

2. запись единицы в знаковый разряд (признак того, что число отрицательное).

Пример. Записать в дополнительном коде двоичные числа

$$X=101001_2 \text{ и } Y = -101001_2.$$

Пусть разрядность ячейки памяти равна 8-ми разрядам. Тогда модуль M будет иметь значение $M=2^7=10000000_2=200_8$.

Очевидно, что $[X]_{\text{д}}=51_8=00101001_2$

Определим $[Y]_{\text{д}}$ (для простоты расчеты будем вести в 8-й системе счисления).

Поскольку Y - отрицательное число:

Шаг 1. Вычисление дополнения: $200_8 - 51_8 = 127_8 = 1010111_2$

Шаг 2. Запись 1 в знаковый разряд. Окончательно $[Y]_{\text{д}}=11010111_2$

Легко убедиться, что $[X]_{\text{д}} + [Y]_{\text{д}} = 0$.

Обратный код

Обратный код $[N]_{\text{обр}}$ целого положительного числа N совпадает с его прямым кодом.

Обратный код целого отрицательного числа получается в результате следующей последовательности действий:

1. инвертирование - замена в цифровой части числа каждой единицы на нуль, а нуля на единицу;

2. запись единицы в знаковый разряд (признак того, что число отрицательное).

Пример. Записать двоичные числа в обратном коде (ячейка 8-ми разрядная):

$$X = 101001. \quad [X]_{\text{обр}} = 00101001$$

$$Y = -101001.$$

Шаг 1. Инвертирование: $101001 \rightarrow 010110$

Шаг 2. Запись 1 в знаковый разряд. Окончательно $[Y]_{\text{обр}} = 1010110$

Из примеров легко видеть, что $[Y]_{\text{д}} = [Y]_{\text{обр}} + 1$.

Из приведенных правил следует, что:

1. Прямой, обратный и дополнительный коды для положительного числа совпадают.

2. В прямом и обратном коде нуль имеет два представления - "положительный" ($00000000 = +0$) и "отрицательный" ($10000000 = -0$) нуль; в дополнительном коде нуль имеет только одно, "положительное" представление;

3. Для получения дополнительного кода отрицательного числа к его обратному коду нужно прибавить единицу;

4. Для преобразования дополнительного кода отрицательного числа в исходное представление (прямой код отрицательного числа) необходимо из дополнительного кода вычесть единицу и инвертировать значащие цифры (разряды).

Алгебраическое сложение. Используя обратный и дополнительный коды, операция алгебраического сложения сводится к арифметическому сложению кодов чисел, которое распространяется и на разряды знаков, которые рассматриваются как разряды целой части числа, при этом отрицательные числа представляются в обратном или дополнительном кодах.

Ограничимся рассмотрением примеров с использованием дополнительных кодов отрицательных чисел. При выполнении операции сложения единица переноса из знакового разряда отбрасывается.

Пример 1. $X = 60_{10} = 74_8 = 111100_2$, $Y = -41_{10} = -51_8 = -101001_2$. В условии $|X| > |Y|$.

Порядок выполнения операции X-Y:

1. Представление первого операнда как положительного числа в прямом коде – $X=00111100$.
2. Представление второго операнда как отрицательного числа в дополнительном коде – $[Y]_д=11010111_2$.
3. Выполнение операции сложения $Z = X + [Y]_д$:

$$\begin{array}{r} 00111100 \\ + 11010111 \\ \hline 00010011 \end{array}$$

Результат: $Z = 00010011_2 = 23_8 = 19_{10}$

Пример 2. $X = 34_{10} = 42_8 = 100010_2$, $Y = -41_{10} = -51_8 = -101001_2$. В условии $|X| < |Y|$.

Порядок выполнения операции X-Y:

1. Представление первого операнда как положительного числа в прямом коде – $X=00100010$.
2. Представление второго операнда как отрицательного числа в дополнительном коде – $[Y]_д=11010111_2$.
3. Выполнение операции сложения $Z = X + [Y]_д$:

$$\begin{array}{r} 00100010 \\ + 11010111 \\ \hline 1111001 \end{array}$$

Результат $[Z]_д = 1111001$. После преобразования получаем

$$Z = 10000111_2 = -7_8 = -7_{10}$$

Пример 3. $X = -14_{10} = -16_8 = -1110_2$, $Y = -41_{10} = -51_8 = -101001_2$.

В условии X и Y – отрицательные числа.

Порядок выполнения операции X-Y:

1. Представление первого операнда как отрицательного числа в дополнительном коде – $[X]_д = 11110010$.
2. Представление второго операнда как отрицательного числа в дополнительном коде – $[Y]_д = 11010111_2$.
3. Выполнение операции сложения $Z = [X]_д + [Y]_д$:

$$\begin{array}{r} 11110010 \\ + 11010111 \\ \hline 11001001 \end{array}$$

Результат $[Z]_д = 11001001$.

После преобразования получаем $Z = 10110111_2 = -67_8 = -55_{10}$

Операции над числами с плавающей запятой. Пусть имеется два нормализованных числа: $X = m_1P^n$ и $Y = m_2P^k$

Тогда, полагая, что $X > Y$, $X + Y = P^n(m_1 + m_2 \cdot P^{(n-k)})$

Аналогично, $X - Y = P^n(m_1 - m_2 \cdot P^{(n-k)})$

Пусть сначала оба числа не отрицательны. Тогда, в соответствии с приведенными формулами, в машине выполняются следующие действия:

1. Происходит выравнивание порядков чисел, причем порядок меньшего числа приводится к порядку большего числа, то есть порядок меньшего числа принимается равным порядку большего числа.
2. Для того, чтобы исходное число не изменило своего значения, мантисса меньшего числа сдвигается на соответствующее число разрядов вправо (уменьшается в соответствующее число раз).
3. Происходит сложение или вычитание мантисс, в результате чего получается мантисса результата. Порядок результата в соответствии с произведенным выравниванием порядков принимается равным порядку большего числа.
4. Полученный результат округляется и, если нужно, нормализуется.

Округление результата может быть реализовано несколькими способами. Например, один из них заключается в том, что предварительный результат вычисляется с количеством разрядов, большим, чем разрядная сетка чисел, над которыми производится операция. Если используется двоичный код, то достаточно одного лишнего разряда. В этом случае округление заключается в прибавлении к этому дополнительному младшему разряду предварительного результата единицы. Если в дополнительном разряде был получен нуль, это не вызовет переноса единицы в основные разряды. Если же в дополнительном разряде записана единица, то предварительный результат будет увеличен на единицу младшего разряда. После проведения этих действий дополнительный разряд отбрасывается.

Таким образом, при выполнении операций сложения и вычитания возникает проблема выравнивания порядков чисел, то есть, прежде всего, должен быть проведен анализ знаков и величин порядков.

Для упрощения операций над порядками в ЭВМ порядок всех чисел увеличивается на одно и то же число так, чтобы порядок наименьшего числа, представимого в машине, был бы неотрицательным, то есть машинный порядок всех чисел n_m удовлетворяет условию

$$n_m = n + K, \text{ где } K = |\max n| + 1$$

Например, если для хранения порядка выделено 6 разрядов, то

$$|\max n| = 11111_2 = 63_{10}, \text{ так что } n_m = n + 64, \text{ то есть } 0 \leq n_m \leq 127.$$

Таким образом, порядки всех чисел в машине становятся положительными, и анализ знака порядка теряет смысл.

Отметим, что при представлении чисел с плавающей запятой отдельно кодируются мантисса и порядок. При этом возможно представление мантисс и порядков чисел в одном и том же или разном кодах. Например, порядок числа может быть представлен в прямом, а мантисса - в дополнительном кодах и т.п.

Таким образом, в общем случае сложение и вычитание чисел с плавающей запятой состоит из следующих этапов:

- сравнение порядков чисел;
- выравнивание порядков чисел;

- перевод мантисс слагаемых в обратный или дополнительный модифицированные коды;
- сложение мантисс в прямом, обратном или дополнительном кодах;
- исправление нормализации вправо или влево;
- перевод мантиссы результата в прямой код;
- выдача результата с учетом порядка.

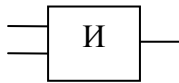
3. БАЗОВЫЕ ФУНКЦИОНАЛЬНЫЕ ЭЛЕМЕНТЫ ЭВМ

3.1. Вентили, последовательностные логические схемы

В вычислительной технике существует понятие логически активных элементов, способных выполнять простейшие действия по преобразованию информации, представленной либо в виде электрических сигналов, либо в виде характеристик намагниченности, света, состояния молекул и т.п. Вне зависимости от физического представления логически активный элемент реализует какую-либо функцию булевой алгебры. Такие элементы называются **вентилями**.

Основу вентиляей составляют крошечные полупроводниковые приборы - транзисторы. Способ объединения транзисторов определяет внутреннюю структуру соответствующего вентиля. Каждый вентиль в соответствии со своей внутренней структурой по-своему изменяет комбинацию импульсов, поступающих на его входы. Три типа логических элементов являются основными (базовыми) - И, ИЛИ, НЕ.

- 1) Вентиль **И**. Реализует логическую операцию **конъюнкции**.
 Условное обозначение Таблица истинности операции



X_1	X_2	$X_1 \wedge X_2$
0	0	0
0	1	0
1	0	0
1	1	1

- 2) Вентиль **ИЛИ**. Реализует логическую операцию **дизъюнкции**.
 Условное обозначение Таблица истинности операции



X_1	X_2	$X_1 \vee X_2$
0	0	0
0	1	1
1	0	1
1	1	1

- 3) Вентиль **НЕ**. Реализует логическую операцию **отрицания**.
 Условное обозначение Таблица истинности операции



X	\bar{X}
0	1
1	0

Последовательностные логические схемы. Последовательностные логические схемы, или схемы с памятью, представляют собой функциональные элементы, имеющие **n** входов и **m** выходов. Значения выходов таких схем в какой-то момент времени являются функциями от значений на входах в этот же момент времени и в некоторое конечное число предыдущих моментов времени. Очевидно, что значения в предыдущие моменты времени должны храниться внутри схемы, то есть некоторым образом запомниться (например, в виде внутренних переменных). Отсюда название - "схемы с памятью".

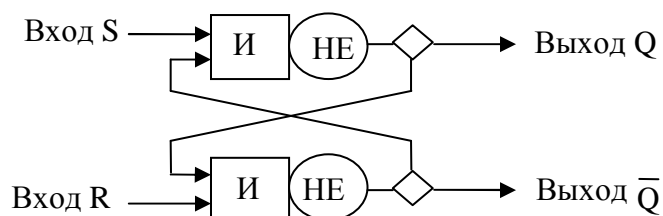
В общем случае состояние схемы меняется каждый раз с появлением новых входных данных. Будем рассматривать схемы, у которых значения выходных переменных в момент времени t зависит от значений входных переменных, поступивших в этот момент, и от внутреннего состояния схемы, сложившегося к этому моменту. Таким образом, предполагается, что схемы срабатывают в дискретные моменты времени по определенным временным тактам.

Очень часто в качестве внутренних переменных, характеризующих состояние системы в момент t_{i+1} используются значения выходных переменных, полученные в момент t_i . Такие схемы называются **схемами с обратными связями**.

Ниже рассматриваются примеры основных схем с памятью, важных для понимания принципов работы аппаратных блоков ЭВМ.

Триггеры. Простейшие последовательностные схемы с обратными связями. Цифровая обратная связь позволяет как бы "удержать" сигнал на выходе после окончания воздействия входного сигнала, то есть придает схеме способность сохранять состояние 0 или 1. Схема, способная хранить ("запоминать") 0 или 1, называется **триггером**.

Триггер-защелка. Работающий по вышеприведенным законам простейший триггер называется базовым или триггером-защелкой (рис. 1.).



S - сигнал установки в единичное состояние,
R - сигнал сброса (установки в нулевое состояние,
Q, \bar{Q} - выходные сигналы (основной и инверсный).

Рис. 1. Триггер-защелка

Если S и R держать в состоянии 1, то триггер не будет менять своего состояния.

Для того, чтобы на выходе Q установить 1, на вход S следует подать 0.

Для того, чтобы сделать \bar{Q} равным 1, на R следует подать 0.

Проанализируем результаты подачи на S и R всех возможных комбинаций значений.

S=0, R=0. На выходе $Q = \bar{Q} = 1$.

S=0, R = 1. На выходе $Q = 1, \bar{Q} = 0$.

S= 1, R= 0. На выходе $Q = 0, \bar{Q} = 1$.

S=1, R =1. Новые значения Q и \bar{Q} совпадают с теми, которые имелись до этого момента.

На основе триггера-защелки строятся триггеры, работающие по другим законам.

Генератор тактовых сигналов (импульсов). Очевидно, что для срабатывания элементарного логического элемента (вентилia или триггера) необходимо некоторое время.

Сигналы 0 или 1 устанавливаются на выходах логических схем через некоторое время I. Пока это время не пройдет, подавать новые сигналы на вход схемы нельзя. Устройство, вырабатывающее синхронизирующие импульсы (синхроимпульсы) в ЭВМ, называется генератором тактовых сигналов.

Функционально синхроимпульс - это единичное значение некоторой входной переменной, причем любая логическая схема может выполнять свою работу только в том случае, если значение этой переменной равно 1. Значение 1 на синхронизирующем входе, логических схем держится в течение отрезка времени, равного длине синхроимпульса (длине такта). К началу следующего синхроимпульса все переходные процессы в электронных схемах должны быть завершены.

Величина такта T выбирается из следующего условия: она должна быть не меньше времени срабатывания самого медленного элемента в цепях логических схем. Реально же в ЭВМ существуют несколько серий синхроимпульсов различной длительности, соизмеримых между собой, которые направляются на разные устройства в соответствии со скоростями их срабатывания.

Синхронизируемые триггеры. Различают два типа входов синхронизируемых триггеров - информационные и синхронизирующие. Сигналы на информационных входах и исходные состояния определяют, каким будет новое состояние триггера. Сигналы на синхронизирующих входах определяют время переключения (синхронизации).

Простейший синхронизируемый триггер называется **RS-триггером** (set, reset - установить, переустановить, рис.2).

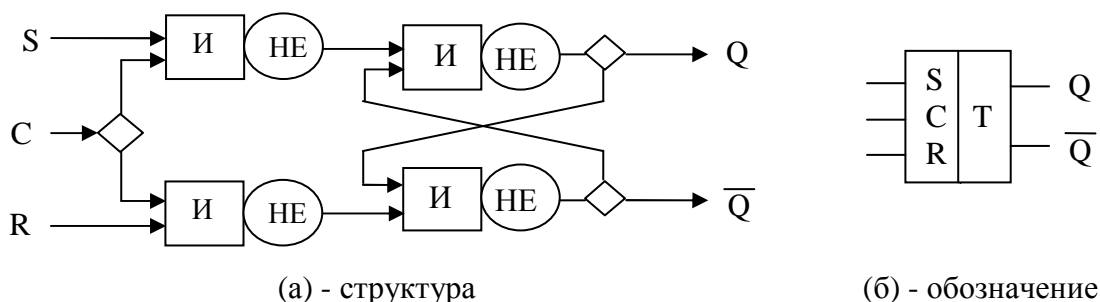


Рис. 2. RS-триггер

Вход S называется входом установки 1, вход R - входом сброса или установки в 0. Вход C является входом синхронизации. Изменить свое состояние синхронный RS-триггер может только в том случае, когда на вход C подан сигнал 1. Если на C -вход подается логический 0, на выходах 1-й и 2-й схем И-НЕ устанавливаются логические 1, и собственно триггер находится в режиме хранения информации. Если же на вход C на короткое время подать логическую 1 (C -импульс), то собственно триггер установится в состояние, зависящее от сигналов на входах S и R . Этим он отличается от триггера-защелки, который может изменить свое состояние в любой момент при подаче сигналов S и R . В следующей таблице отражены все возможные варианты переходов (состояний) RS-триггера.

Таблица состояний RS-триггера

C	S	R	Q_{t+1}	\bar{Q}_{t+1}	Пояснения
1	0	0	Q_t	\bar{Q}_t	Режим хранения информации $S=R=0$ или $S=R=1$
1	0	1	0	1	Режим установки 0 ($Q_t=0$ или $Q_t=1$)
1	1	0	1	0	Режим установки 1 ($Q_t=0$ или $Q_t=1$)
1	1	1	неопр.	неопр.	Запрещенная комбинация
0	x	x	Q_t	\bar{Q}_t	

Здесь Q_t - состояние триггера в текущий временной такт, Q_{t+1} - состояние триггера в последующий временной такт.

Как правило, рассматриваются синхронные (или тактируемые) схемы, которые меняют свое состояние только в том случае, когда переменная синхронизации равна 1. Такие моменты времени называются **тактами**.

Состояние триггера при наличии логической 1 на обоих входах не определено, потому что бессмысленно требовать установки и сброса триггера одновременно. Состояние реального триггера при таких входных сигналах зависит от его устройства.

Триггер задержки. Триггер задержки или D-триггер (delay - задержка) имеет только один информационный вход (рис. 3.).

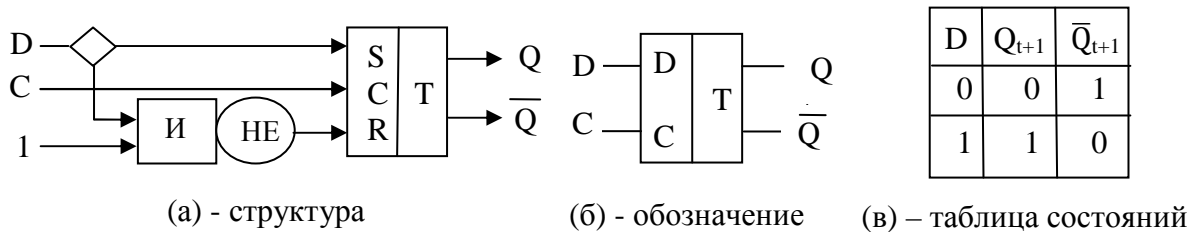


Рис. 3. D-триггер

Легко убедиться, что после С-импульса триггер повторяет сигнал на входе D. D-триггер выполняет только функции хранения поданного на его входы значения. И это значение может быть считано (скопировано) сколько угодно раз, пока его не сменит новое значение.

Таким образом, триггерная схема имеет два устойчивых состояния и является, во-первых, запоминающим устройством на один разряд, и, во-вторых, логическим устройством, меняющим свое состояние в зависимости от проходящих на вход сигналов.

T-триггер. Простой T-триггер с одним (счетным) входом функционирует следующим образом: с приходом каждого нового сигнала, равного 1, триггер меняет свое внутреннее состояние на противоположное (рис.4.).

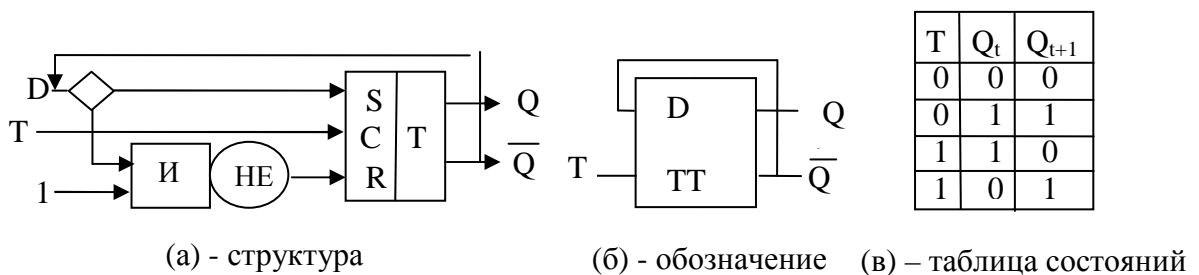


Рис.4. T-триггер

Регистры и счетчики. Триггеры - это элементы, на основе которых строятся регистры различного назначения. Перечислим некоторые из этих устройств.

1) **Сдвиговой регистр.** Регистр с последовательной записью и последовательным считыванием информации. При каждом импульсе управления весь код (содержимое) регистра смещается по отношению к цепочке триггеров на одну позицию влево или вправо. На рис. 5. изображена структура и схема функционирования сдвигового регистра влево.

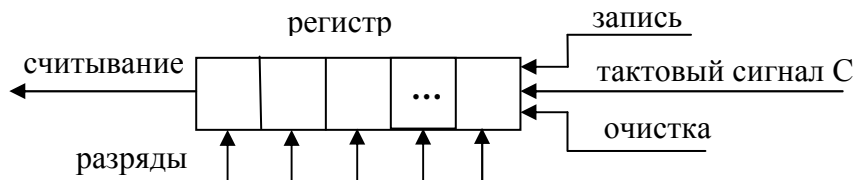


Рис. 5. Сдвиговой регистр

Сдвиговой регистр функционирует по следующему алгоритму. В такт t+1 из каждого разряда 1 регистра в соседний слева (старший) разряд поступает соответствующее значение, то есть вся информация, расположенная в разрядах регистра, сдвигается на один разряд. При этом младший разряд устанавливается в состояние, соответствующее сигналу, появившемуся на единственном входе регистра. Очевидно, что регистр с последовательной записью и последовательным считыванием может быть реализован на основе d-триггеров.

Заметим, что сдвиг влево содержимого регистра, трактуемого как число, эквивалентен умножению исходного числа на 2, а сдвиг вправо - делению этого числа на 2.

2) **Регистр с параллельной записью и параллельным считыванием.** Такой регистр может быть реализован на RS- и D-триггерах; схема регистра изображена на рис. 6.

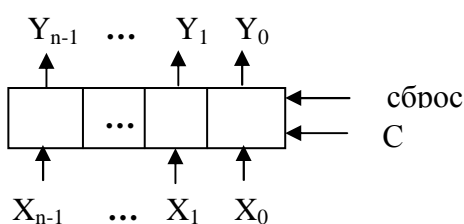


Рис. 6. Схема регистра с параллельной записью и параллельным считыванием

3) **Регистр-счетчик.** На каждом такте при подаче на вход регистра-счетчика (или суммирующего счетчика) 1 величина хранимого в нем числа увеличивается на 1. Схема регистра изображена на рис. 7.

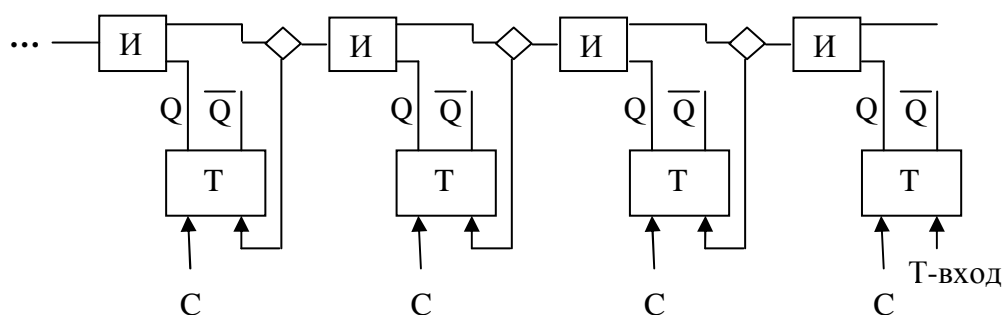


Рис. 7. Схема регистра-счетчика

Функционирует регистр-счетчик следующим образом. На Т-вход младшего разряда регистра в каждый такт поступает сигнал 0 или 1. Сигналы 1 пересчитываются с использованием Т-триггеров, которые с приходом 1 меняют значение (внутреннее состояние) на противоположное. Схемы И управляют переносами из разряда в разряд. Заметим, что каждый Т-триггер соединяется со схемой И выходом со старым значением своего внутреннего состояния.

4) **Инвертируемый регистр.** Если разряды регистра реализованы с использованием Т-триггеров, то, направляя параллельно на все Т-входы разрядов единичный сигнал, можно значения разрядов инвертировать.

Схемы регистров, используемых в современных ЭВМ, часто многофункциональны. Так, например, для выполнения операций часто значения операндов предварительно должны быть преобразованы в требуемую форму. Для операция вычитания требуется, чтобы один из операндов был инвертирован. Для реализации арифметических действий над числами с плавающей запятой необходимо выравнивание порядков, то есть выполнение операции сдвига на определенное количество разрядов, для чего необходимо иметь счетчик сдвигов, и так далее. Вид функции, которую должна выполнять схема такого многофункционального регистра, выбирается на основе комбинации значений сигналов управления. Эти сигналы вырабатываются устройством управления в процессе расшифровки команд программы.

3.2. Комбинационные схемы, понятие об интегральной схеме

Кроме регистров в состав ЭВМ входят схемы для выполнения множества разнообразных функций. Так например, в состав арифметико-логического устройства, входящего в состав центрального процессора, должны быть включены схемы реализации операций сложения, вычитания двоичных чисел, в состав устройства управления - схемы

дешифровки кода операции, заданной в команде, и т.д. Все эти схемы относятся к, так называемым, **комбинационным схемам**. В отличие от схем с памятью, значения переменных на выходе комбинационной логической схемы однозначно зависят только от значения входных переменных. Далее обсуждается несколько комбинационных схем, занимающих важное место в функционировании ЭВМ.

Полусумматор. На рис. 8. представлена схема сложения одноразрядных двоичных чисел, которую в своей работе может использовать арифметико-логическое устройство.

Обозначения:

X и Y - входные переменные схемы,

S - выходная переменная соответствующая сумме,

C - выходная переменная, определяющая возможный перенос (переполнение).

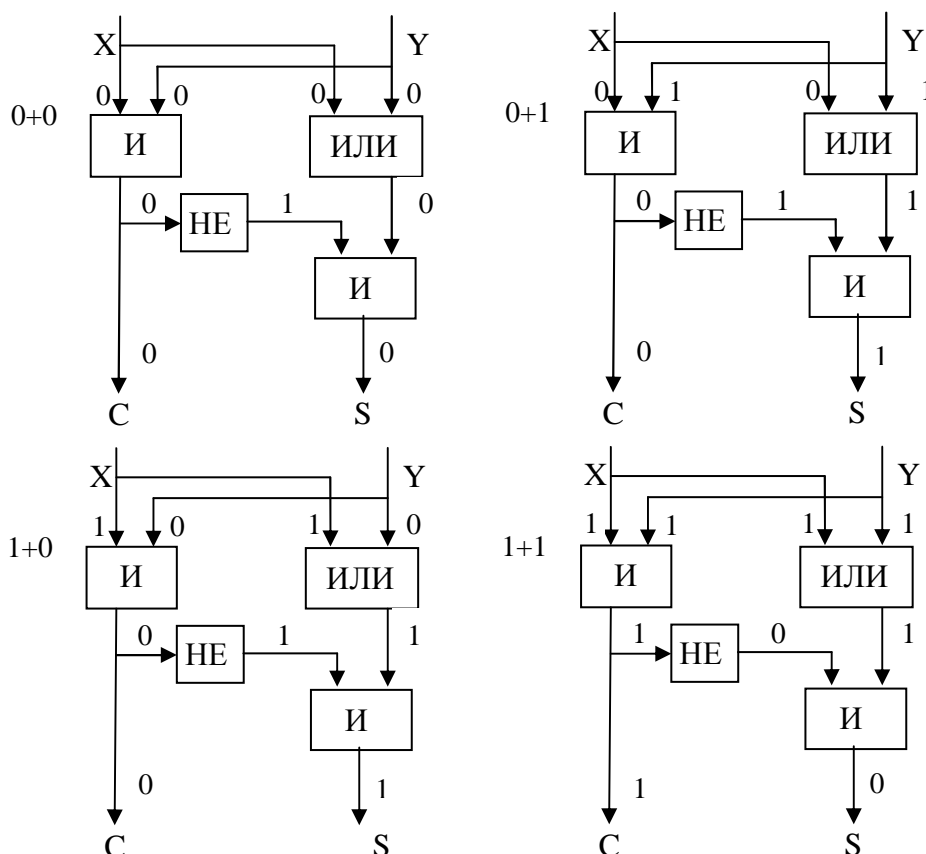


Рис. 8. Схема сложения одноразрядных двоичных чисел в полусумматоре

Полный одноразрядный сумматор. Полусумматор, дополненный третьим входом - значением разряда переноса из соседнего младшего разряда (рис. 9.).

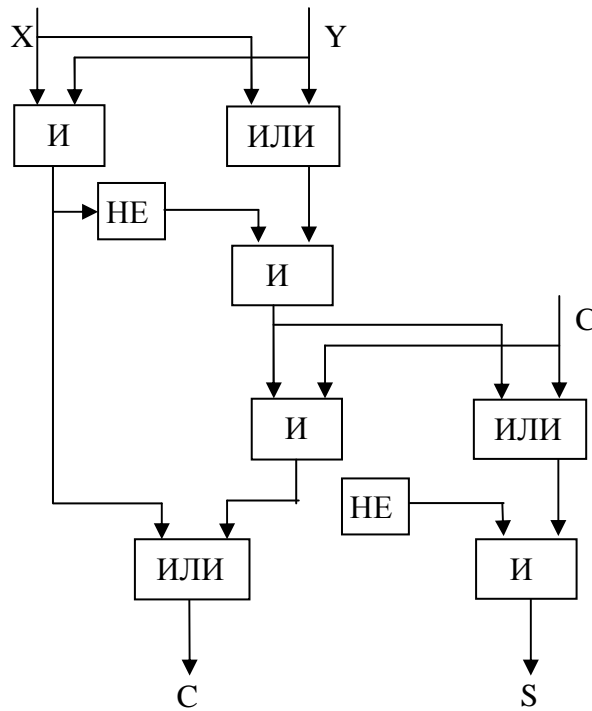


Рис. 9. Схема полного одноразрядного сумматора

Сумматор. Многоразрядный сумматор получается последовательным соединением одноразрядных сумматоров (рис. 10.).

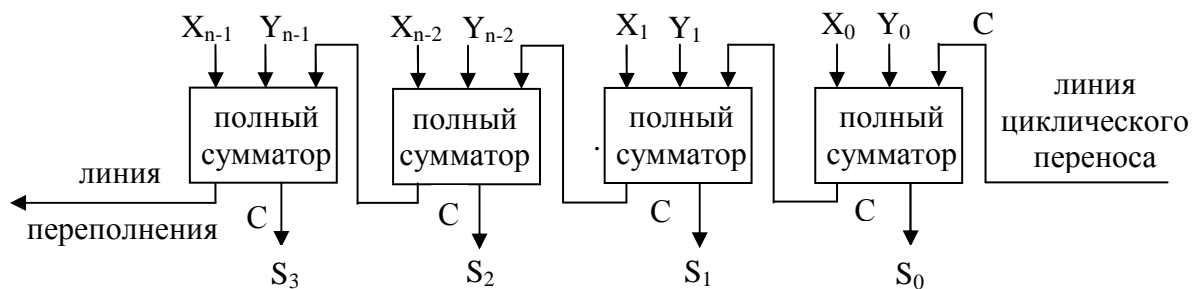


Рис. 10. Схема многоразрядного сумматора

Появление 1 на линии переполнения указывает на возникновение переноса из старшего разряда при сложении двух целых чисел (ситуация переполнения). При соединении линии переполнения с линией циклического переноса сумматор может складывать числа, представленные в обратном коде. При сложении чисел в прямом коде по линии циклического переноса в полный сумматор младшего разряда должен быть подан 0.

Шифратор. Это устройство преобразует управляющий сигнал в двоичный код (рис. 11.).

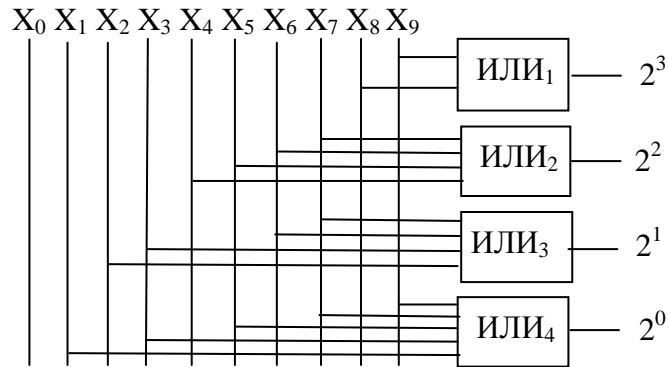


Рис. 11. Структура двоичного шифратора

Логическая 1 на любой из вертикальных шин преобразуется в параллельный двоичный код. Например, 1 на линии X_9 преобразуется в двоичный код

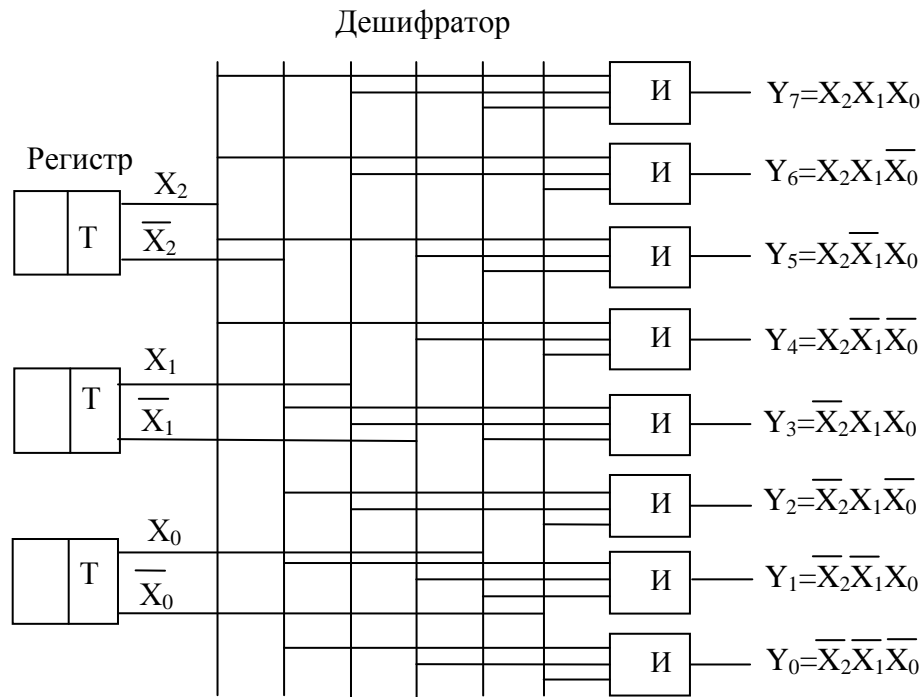
$$1001 = (\text{ИЛИ}_1 + \text{ИЛИ}_4) = 2^3 + 2^0 = 9.$$

Дешифратор. Комбинационная схема, имеющая n -входов и 2^n выходов. Реализует функцию, обратную шифрованию. При любой комбинации входных сигналов X сигнал 1 наблюдается только на одном из выходов Y , причем комбинация значений входных сигналов рассматривается как двоичное число, определяющее номер выходного сигнала Y , который требуется установить в 1. Значения остальных выходных сигналов должны быть установлены в 0. На рис.12. изображена схема простейшего трехразрядного двоичного дешифратора.

На схеме вход X_0 соответствует 2^0 , вход X_1 - 2^1 и X_2 - 2^2 . Эта схема может рассматриваться как преобразователь двоичного натурального кода в код "1 из n ". Дешифраторы широко используются в схемах управления памятью. На вход дешифратора подается адрес (целое двоичное число), определяющий номер ячейки памяти, к которой требуется обращение. Это число и представляет набор значений входных переменных дешифратора. В результате на выходе дешифратора возбуждается только одна выходная линия, по которой в свою очередь инициируется работа схем управления запрошенной ячейки.

№	X2	X1	X0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

(а) Таблица состояний



(б) Структура и принцип подключения к регистру

Рис. 12. Структура простейшего дешифратора

Мультиплексор. Устройство сведения информации в одну шину. Имеет 2^n сигнальных входов X и n селектирующих (управляющих) входов S и только один выход Y . Единственный выход мультиплексора принимает значение той входной сигнальной переменной, номер которой указан набором селектирующих сигналов. Таким образом, если набор, поступающий по селективным линиям, представляет двоичное число k , то на единственном выходе появится значение входной переменной $Y=X_k$.

Демультимплексор. Устройство, производящее разделение информации по адресам назначения. Схема имеет один вход X , n селектирующих входов S и 2^n выходов Y . Действие демультимплексора прямо противоположно действию мультиплексора. Сигнал X направляется на тот выход, номер которого указан в селектирующем наборе переменных.

Реализовать как мультиплексор, так и демультимплексор можно, используя дешифратор. На рис. 13. поясняется принцип действия схем. Предполагается, что исходные состояния счетчиков одинаковы, импульсы управления строго синхронны.

Синхронно на выходе обоих счетчиков последовательно под влиянием входных импульсов формируются целые двухзначные двоичные числа, соответствующие десятичным числам 0, 1, 2, 3. Под воздействием комбинаций селективных сигналов S_0 и S_1 на выход мультиплексора последовательно будут поданы значения входных переменных X_0 , X_1 , X_2 и X_3 . Очевидно, что на выходе дешифратора через четыре такта будет построен код, совпадающий с входным кодом мультиплексора.

Интегральная схема (микросхема). Комбинационная или последовательная схема, реализованная на одном срезе кристалла кремния на основе сложного и высокоточного технологического процесса. Этот процесс включает в себя создание логически активных элементов путем внедрения в соответствующие точки кремниевой пластинки окислов редкоземельных элементов и напыления металлических площадок. Таким путем в едином технологическом процессе удастся создать на поверхности кристалла диоды, транзисторы и проводники, связывающие отдельные вентили в единое целое.

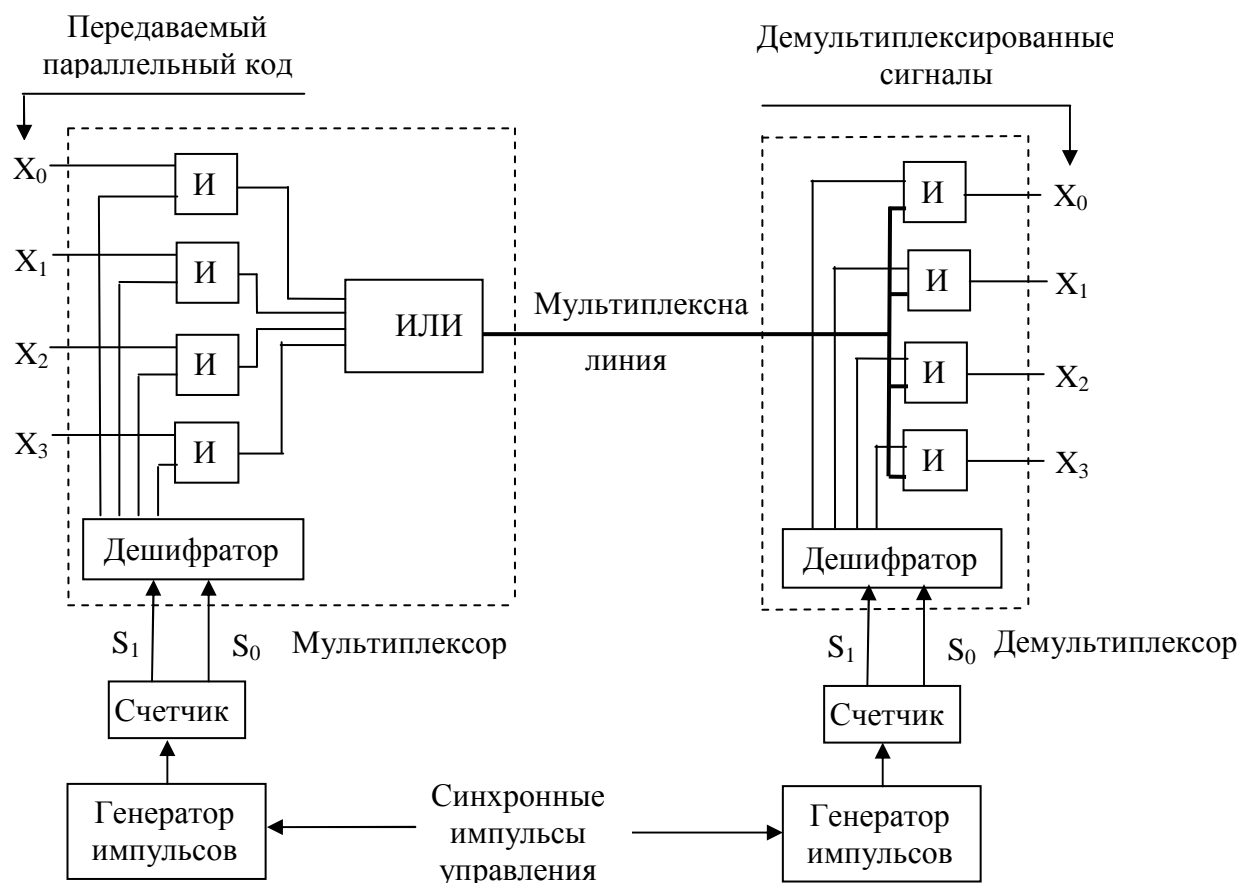


Рис. 13. Мультиплексор-демультиплексор

4. ОБЩАЯ ОРГАНИЗАЦИЯ ЭВМ

4.1. Понятие архитектуры ЭВМ

В широком смысле архитектурой ЭВМ называют совокупность ее свойств и характеристик, рассматриваемую с точки зрения пользователя.

Под вычислительной системой (ВС) будем понимать устройство, воспринимающее информацию в виде данных, представленных в цифровой (дискретной) форме, обеспечивающее хранение данных, их переработку с большой скоростью, пересылающее данные внутри системы по линиям связи и выдающее результат этих действий как информацию, то есть в требуемой для пользователя форме. Заметим, что в отличие от цифровых, в аналоговых системах (системах непрерывного действия) данные представляются физически непрерывными величинами - чаще всего величиной электрического напряжения. Вычислительные системы, объединяющие в своем составе как цифровые, так и аналоговые устройства, называются гибридными.

Пользователя вычислительной системы в первую очередь интересует ее производительность на классе задач, для решения которых вычислительная система приобретается; эта производительность, в свою очередь, во многом зависит от системы программирования (применяемого инструментария - алгоритмического языка - для записи алгоритма), операционной системы (системного программного обеспечения), состава внешних (периферийных) устройств и способов обращения к ним.

Архитектурное описание вычислительной системы включает описание ее структуры и характеристик всех ресурсов (аппаратных и программных), доступных пользователю. В свою очередь структура вычислительной системы определяется описанием состава ее

функциональных устройств, связей и принципов взаимодействия блоков. В данном пособии обсуждается только аппаратный уровень вычислительной системы - структура ЭВМ и характеристики аппаратных ресурсов.

4.2 Обобщенная структурно-функциональная схема ЭВМ

Решение задач на ЭВМ (вычислительный процесс) проводится по определенным правилам. Кроме *исходных данных*, вводимых в ЭВМ, для решения задачи необходим *алгоритм*, то есть совокупность точных предписаний (правил), определяющих вычислительный процесс, например, алгоритм расчета по алгебраической формуле должен указывать на правильный порядок выполнения в формуле арифметических операций. Для выполнения на ЭВМ алгоритм решения задачи должен быть представлен в виде *программы* - описания алгоритма на каком-либо понятном ВС формализованном языке. Эта программа определяет действия машины и последовательность их выполнения при обработке исходных данных и промежуточных результатов.

Для выполнения процесса вычислений в состав ЭВМ входит ряд устройств, каждое из которых выполняет вполне определенные законченные функции, то есть является функционально-законченной частью технических средств ЭВМ.

Базовая структура современных серийных ЭВМ, по существу, аналогичны структуре и основам функционирования своих предков - вычислительных машин первых поколений, сформулированных [фон-Нейманом](#). На рис. 14. представлена обобщенная схема типичной ЭВМ.

Основой ЭВМ является **процессор**, непосредственно осуществляющий обработку данных и управление этим процессом; в состав процессора входят:

- устройство управления выборкой команд из памяти и их выполнением;
- арифметико-логическое устройство, производящее операции над данными;
- регистры, осуществляющие временное хранение данных и состояний процессора;
- схемы для управления и связи с подсистемами памяти и ввода-вывода.

Следует заметить, что наряду с процессором, осуществляющим общее управление функционированием вычислительной системы и выполнением программ (**центральным процессором**), в состав современных ЭВМ могут входить специализированные процессоры (например, **графический процессор**). Термин процессор в данном пособии обозначает только [центральный процессор](#).

Оперативная память обеспечивает хранение данных и программ, обрабатываемых процессором, и подробно рассматривается в [п.6.2.1](#). Здесь для простоты будем считать, что процессор обрабатывает информацию (команды и данные), располагающиеся только в оперативной и ни в каких других видах памяти (например, внешней).

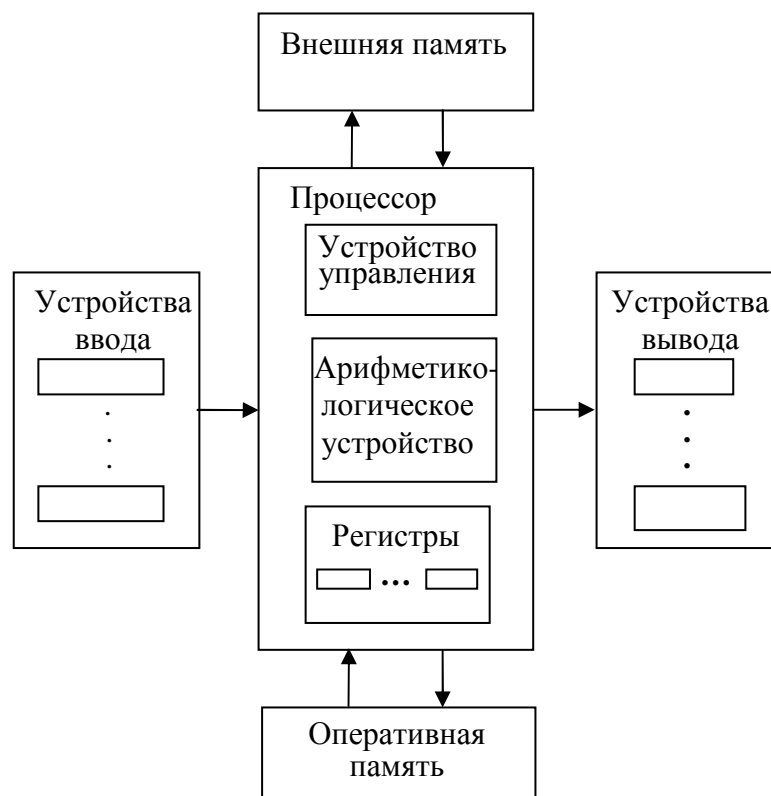


Рис. 14. Обобщенная структурная схема ЭВМ

Устройства ввода обеспечивают считывание информации (программы решения задачи и исходных данных) с определенных носителей информации (клавиатур, датчиков состояний управляемых объектов и т.п.) и ее представление в форме электрических сигналов, воспринимаемых другими устройствами ЭВМ (процессором или основной памятью).

Устройства вывода представляют результаты обработки информации в форме, удобной для визуального восприятия (экран дисплея, печатающие устройства, графопостроители, и т.п.). При необходимости они обеспечивают запоминание результатов на носителях, с которых эти результаты могут быть снова введены в ЭВМ для дальнейшей обработки (например, флеш-карты), или передачу результатов на исполнительные органы управления, например, робота.

Внешняя память обеспечивает возможность накопления библиотек программ и данных. Внешнюю память ЭВМ образуют устройства, способные обеспечить размещение и длительное хранение большого объема информации с достаточно быстрым доступом к ней (магнитные диски, магнитные ленты, и т.п.).

Замечание. Всю совокупность (парк) устройств ввода, вывода и внешней памяти называют **внешними устройствами (ВУ)**.

Устройство управления предназначено для автоматического выполнения программ в ЭВМ. **Центральное устройство управления** (устройство управления центрального процессора или в дальнейшем просто **устройство управления**) вырабатывает управляющие сигналы, необходимые для выполнения всех операций, предусмотренных системой команд, а также координирует работу всех узлов и блоков ЭВМ. В целом устройство управления реализует следующие основные функции:

- производит выборку очередной команды программы;
- вырабатывает последовательность управляющих сигналов для выполнения операции, заданной в команде;
- определяет адрес следующей команды программы, которую необходимо выбрать и выполнить на очередном шаге работы ЭВМ.

Перечисленные функции определяют самый низкий уровень управления процессами в вычислительной системе. На более высоких уровнях блоки управления осуществляют взаимодействие между различными подсистемами ЭВМ и синхронизацию их работы.

Стрелки \longrightarrow и \longleftarrow на этой схеме указывают на возможные типы операций обмена (перемещения данных между устройствами), выполняемых процессором. Возможны два типа таких операций – **ввод** (или **чтение**) и **вывод** (или **запись**). Важно различать пары взаимодействующих устройств:

1. Процессор – основная память:
 - чтение – перенесение (выборка) информации из памяти в процессор;
 - запись – перенесение данных из процессора в память.
2. Основная память – устройства ввода-вывода:
 - чтение – перенесение (выборка) данных из УВВ в основную память;
 - запись – перенесение данных из основной памяти в УВВ.

5. ПРОЦЕССОР

5.1. Команда ЭВМ

Формат машинной команды – это одна из основных характеристик архитектуры центрального процессора ЭВМ. В общем виде структура команда включает два поля:

Код операции	Поле операндов
--------------	----------------

1. Код операции (КОП) указывает операцию, которую должна выполнить ЭВМ (сложение, вычитание, сравнение, переход и т.п.).
2. В поле операндов хранятся указатели на местоположение операндов (адреса памяти или номера (адреса) регистров); в частном случае в качестве операндов могут указываться и непосредственные значения констант.

Заметим, что в простейшем случае поле операндов может отсутствовать (например, команда ОСТАНОВ); такие команды носят название **безадресных**.

Ниже приводятся возможные форматы **адресных** команд (с непустым полем операндов).

Замечание. В состав системы команд процессора входят команды, операндами которых не являются адреса. Принципы функционирования таких команд здесь не рассматриваются.

Одноадресная команда. Структура поля операндов

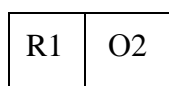


Схема выполнения:

1. выполняется операция, первым операндом которой является содержимое внутреннего хранилища (регистра) процессора - **аккумулятора** (часто называемого **сумматором**), вторым - содержимое ячейки памяти по адресу А;
2. результат заносится в аккумулятор.

Очевидно, что, если одноадресная команда является командой перехода, адрес указывает на команду, которая изменяет последовательный ход выполнения программы.

Двухадресная команда. Структура поля операндов



R1 – адрес (номер) регистра процессора,

O2 – адрес регистра процессора (R2) или ячейки памяти (A2).

Схема выполнения:

1. выполняется операция над операндами по адресам R1 и O2;
2. результат заносится по адресу R1 (уничтожая значение первого операнда).

Замечание. В современных универсальных ЭВМ двухадресный формат является основным.

Трехадресная команда. Структура поля операндов

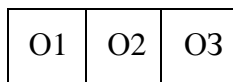


Схема выполнения:

1. выполняется операция над операндами по адресам O1 и O2;
2. результат заносится по адресу O3.

Четырехадресная команда. Структура поля операндов

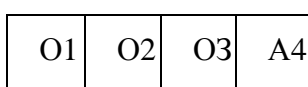


Схема выполнения:

1. выполняется операция над операндами по адресам O1 и O2;
2. результат заносится по адресу O3;
3. следующей выполняется команда, расположенная в памяти по адресу A4.

5.2. Общая структура процессора

Основной состав современного процессора включает:

- **арифметико-логическое устройство - АЛУ**, производящее операции и вырабатывающее ряд признаков (результат меньше нуля, равен нулю или больше нуля и т.п.);
- **регистр команд**, служащий для размещения исполняемой (текущей) команды;
- **счетчик команд**, указывающий на адрес следующей команды, которая должна быть выполнена после завершения выполнения текущей команды; содержимое счетчика команд увеличивается в момент выборки из памяти текущей исполняемой команды на ее длину или, если выбрана команда перехода, может быть заменено на адрес, указанный в адресной части команды перехода;
- **регистр адреса**, содержащий адрес ячейки памяти, из которой будет считана команда (или операнд) или в которую требуется записать результат выполнения команды;
- **сумматор**, используемый для размещения результата выполнения операции;
- **регистр флагов**, хранящий признаки результата (флаги) выполнения последней операции, его вырабатывающей; эти признаки используются для организации работы команд переходов;
- совокупность **регистров общего назначения (РОНы)** – внутренняя (локальная) память процессора.

Замечание. В приведенном выше формате двухадресной команды первый (R1) и второй (R2) операнды – это номера РОНов.

В простейшем случае для решения задачи на ЭВМ необходимо:

1. через устройство ввода информации загрузить (перенести) в оперативную память программу решения задачи и исходные данные;

2. "сообщить" процессору адрес ячейки памяти, в которой размещена первая команда программы; подобное "сообщение" реализуется занесением искомого адреса в счетчик команд;
3. инициировать процедуру передачи в подсистему памяти адреса первой команды программы и пересылки ее содержимого из памяти в регистр команд; с этого момента процессор начинает выполнять последовательность указанных в программе операций.

Алгоритм выполнения команд простого процессора может быть представлен схемой на рис. 16.

Как видно из схемы алгоритма, два первых действия (блоки 1 и 2) выполняются для каждой команды. Эти действия, а также действия по определению типа команды - блоки 3, 6 и 9, обычно называют "Выборкой команды". Последующие действия алгоритма ("Исполнение команды") полностью зависят от того, какая это команда.

Замечание. В блоке 9 определяется два типа команд – пересылка из регистра в память (запись в память, выгрузка) или пересылка из памяти в регистр (чтение, выгрузка).

Упрощенная схема взаимодействия компонентов процессора при выполнении двухадресной команды представлена на рис.17.

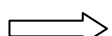
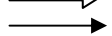
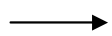
На схеме приняты следующие условные обозначения.

Системная магистраль. В данном пособии рассматриваются основы организации ЭВМ с *магистральной архитектурой*. Системная магистраль обеспечивает коммуникационную среду взаимодействия всех устройств ЭВМ с процессором. В частности, к системной магистрали присоединяется и подсистема управления оперативной памятью (оперативное запоминающее устройство – ОЗУ). Системная магистраль обеспечивает перемещение между процессором и устройствами информации трех видов: адреса, данные, сигналы управления.

Внутренняя магистраль. Коммуникационная среда взаимодействия внутренних компонентов процессора.

Интерфейсный блок. Осуществляет перемещение информации всех видов между системной и внутренней магистралями.

Стрелки:

-  – перемещение данных,
-  – перемещение адреса,
-  – перемещение сигналов управления.

БР1 – буферный регистр первого операнда (R1).

БР2 – буферный регистр второго операнда (O2).

1, 2, 3, ..., 12 – последовательность действий при выполнении команды.

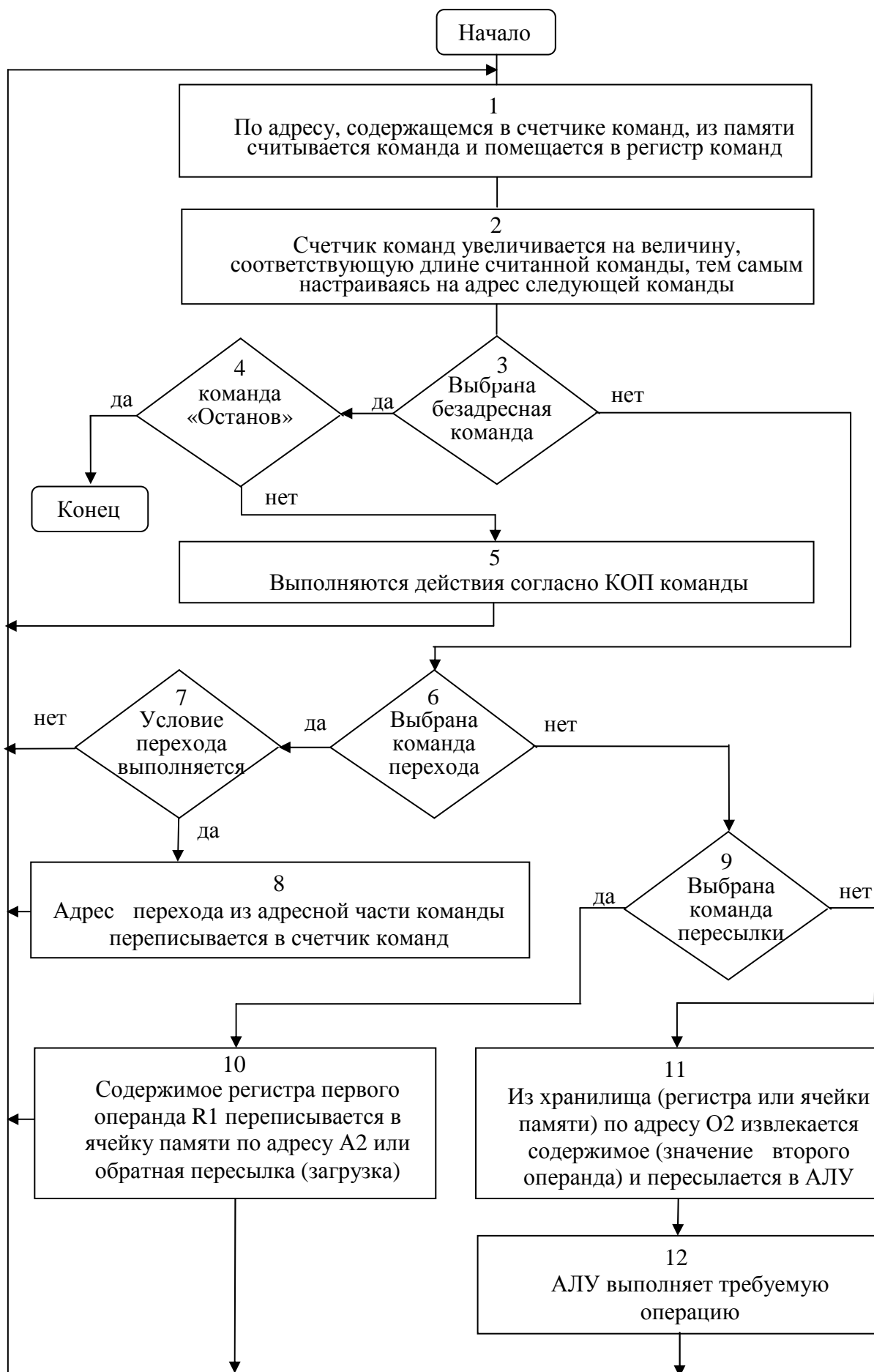


Рис. 16. Схема алгоритма выполнения команд простого процессора

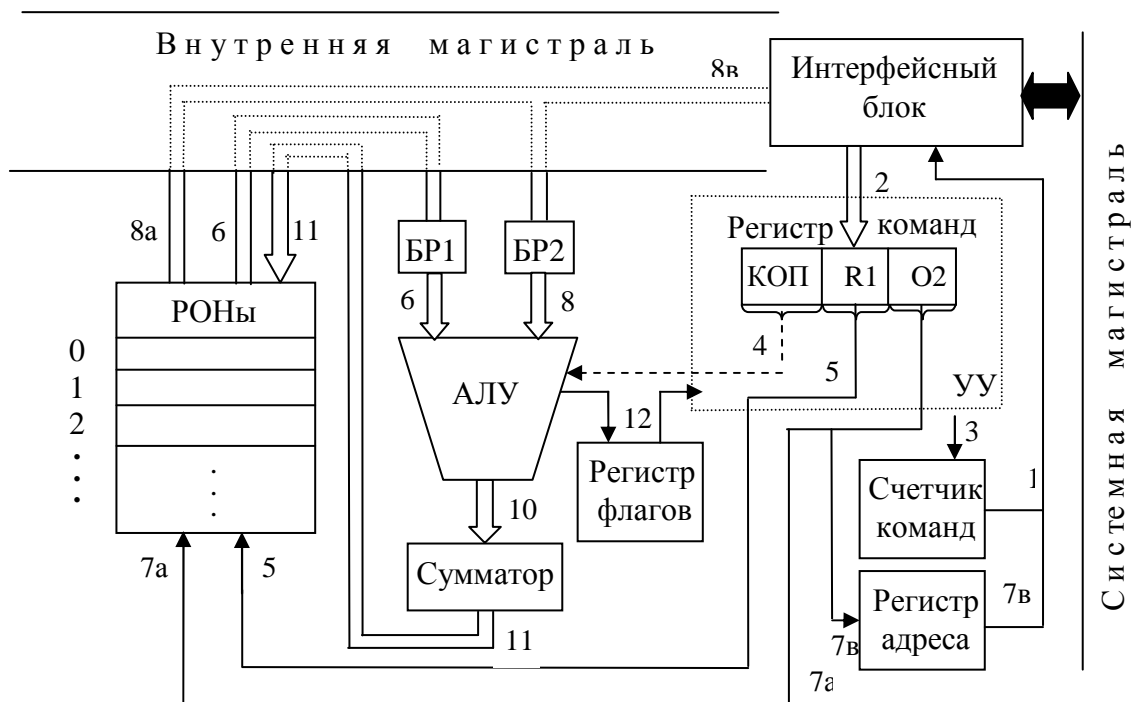


Рис. 17. Упрощенная схема выполнения машинной команды

Рассмотрим искомую последовательность действий подробнее.

1. Адрес команды, которая должна быть выполнена, пересылается из Счетчика команд в ОЗУ.
2. Выбранная из ОП команда загружается в Регистр команд.
3. Устройство управления (УУ) определяет длину команды и увеличивает значение Счетчика команд, настраивая его содержимое на адрес в ОП следующей команды, которая должна быть выполнена.
4. УУ расшифровывает КОП выполняемой команды, формирует сигналы управления и пересылает их в АЛУ.
5. УУ расшифровывает адрес первого операнда (номер РОН R1) и формирует управляющие сигналы на выборку содержимого соответствующего регистра.
6. Содержимое регистра первого операнда по внутренней магистрали пересылается в БР1.
7. УУ расшифровывает адрес второго операнда O2. Возможны два варианта.

Вариант 1.

- 7а. Второй операнд – номер РОН. УУ формирует управляющие сигналы на выборку содержимого соответствующего регистра.
- 8а. Содержимое регистра второго операнда пересылается во внутреннюю магистраль.

Вариант 2.

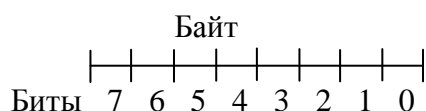
- 7в. Второй операнд – адрес ОП. УУ пересылает искомый адрес в Регистр адреса, после чего следует запрос к ОЗУ на выборку содержимого требуемой ячейки ОП.
- 8в. Содержимое ячейки ОП пересылается по системной магистрали во внутреннюю магистраль процессора.
8. Значение второго операнда пересылается в БР2.
9. АЛУ выполняет требуемую операцию.
10. Результат реализованной операции поступает в Сумматор.
11. Содержимое Сумматора пересылается в регистр R1.
12. Признаки результата, сформированные АЛУ, пересылаются в регистр флагов (регистр состояний) и из него – в УУ.

Очевидно, что данная схема предельно упрощена и дает самые общие представления о функционировании универсального центрального процессора.

6. ПОДСИСТЕМА ПАМЯТИ

6.1. Структура и характеристики памяти

Структура памяти. Память состоит из огромного числа **элементов** памяти, каждый из которых может находиться в одном из двух состояний, кодируемых двоичной цифрой 0 или 1 (**биты**, кодирующие состояния элементов). Биты памяти группируются в более крупные единицы хранения информации, минимальной из которых является **байт**. Вне зависимости от модели ЭВМ длина байта информации фиксирована и составляет 8 бит (двоичных разрядов).



Замечание. Одной из характеристик ЭВМ является **длина машинного слова**. В различных моделях ЭВМ (точнее, процессорах) размер машинного слова различен (8 бит, 16 бит, 32 бита, 64 бита) и зависит от соответствующей характеристики шины данных системной магистрали.

Модули памяти (хранилища информации) различного типа, объединяясь с соответствующими блоками управления, образуют класс **запоминающих устройств** (ЗУ), например, оперативная память (ОП) и оперативное запоминающее устройство (ОЗУ). Часто, однако, понятия памяти и запоминающего устройства используют как синонимы.

Ниже перечисляются характеристики памяти и основные параметры, определяющие различные типы памяти.

Информационная емкость. Информационная емкость (или просто **емкость**) памяти выражается в количестве битов, байтов или слов; так как эта емкость может быть очень велика, обычно используют более крупные единицы, образованные присоединением приставок **К** (кило - $2^{10}=1024$), **М** (мега - $2^{20}=1048676$), **Г** (гига - $2^{30}=1073741824$), **Т** (тера - $2^{40}=1\ 099\ 511\ 627\ 776$) к перечисленным единицам.

Операции с памятью. В отличие от других функциональных схем, занимающихся переработкой и преобразованием информации, запоминающие устройства выполняют только одну функцию - хранение информации. Поэтому к запоминающим устройствам могут быть применены только две операции - **ЧТЕНИЕ (ВЫБОРКА)** и **ЗАПИСЬ**.

В любой момент, обратившись к памяти, можно получить содержимое хранимого там слова, то есть реализовать операцию **ЧТЕНИЯ** из памяти. Большинство запоминающих устройств устроены таким образом, что процесс выборки не меняет их содержимого - они как бы выдают во вне копию информации, которая в них продолжает храниться. Таким образом, один раз записав слово, можно сколько угодно раз получать его копии.

При операции **ЗАПИСИ** в память старое содержимое безвозвратно теряется и замещается новой информацией.

Цикл считывания и цикл записи определяются, соответственно, как время от момента выдачи процессором сигнала на считывание или запись и поступления информации, необходимой для выполнения операции, до того момента, когда заканчиваются все действия, связанные с выполняемой операцией, и память будет готова реализовать следующую операцию. Иногда, при совпадении продолжительности этих циклов, используют обобщенный термин - **цикл памяти**.

Способ выборки информации. Различают два основных типа запоминающих устройств: с **произвольной выборкой** и **последовательной выборкой**.

При произвольной выборке время доступа к заданному слову не зависит от местонахождения этого слова в памяти, при последовательной - зависит.

Методы размещения и поиска информации. По этим признакам различают следующие виды основной памяти ЭВМ: *адресная, стековая и ассоциативная*.

Доступ к информации в **адресной** памяти производится по адресу ячейки, начиная с которой информация в памяти располагается (очевидно, что в общем случае информационный массив может занимать несколько ячеек).

При размещении информации в **стековой** памяти возможен доступ только к ее верхушке, хотя на практике это требование может быть смягчено.

Поиск информации в **ассоциативной** памяти производится по содержимому ее ячеек.

Возможность изменения информации. Существуют элементы памяти с легко изменяемыми состояниями, способные работать как в режиме чтения, так и в режиме записи. Есть более дешевые элементы памяти, которые, единожды установленные в 0 или 1, изменить свое состояние при функционировании ЭВМ не могут. Запоминающие устройства, построенные на таких элементах, используются только в режиме чтения.

Сохранение информации при отключении питания ЭВМ. В некоторых видах запоминающих устройств происходит потеря информации при отключении (и даже кратковременном прерывании) питающего напряжения. Такие запоминающие устройства называют *энергозависимыми*. В *энергонезависимых* запоминающих устройствах (например, магнитной памяти) информация сохраняется.

6.2. Классы запоминающих устройств

Общая классификация. По своему *функциональному* назначению запоминающие устройства можно разделить на три крупных класса:

- **регистровые** внутренние запоминающие устройства;
- **Кэш-память;**
- **основная память;**
- **внешние** запоминающие устройства (ВЗУ).

Регистровые запоминающие устройства входят в состав центрального процессора, служат для промежуточного хранения данных, адресной и управляющей информации. Эти запоминающие устройства обладают наибольшим быстродействием и обычно невелики по объему. Совокупность РОНов, образующих *локальную* память процессора, предназначена для повышения быстродействия процессора, поскольку загрузка в АЛУ операндов выполняемой команды из РОНов гораздо быстрее, чем из оперативной памяти.

Сверхоперативная (или **КЭШ-память**) используется как буферная память, сглаживающая различие в быстродействии процессора и оперативной памяти.

Запоминающие устройства, входящие в состав **основной памяти**, составляют важнейший модуль любой ЭВМ, в них хранятся программы и данные, обрабатываемые центральным процессором. В составе основной памяти выделим **оперативное** запоминающее устройство (ОЗУ) и **постоянное** запоминающее устройство (ПЗУ).

К классу **внешних запоминающих устройств** (ВЗУ) относятся энергонезависимые устройства с памятью (магнитные и оптические диски, флеш-карты, магнитные ленты). Существенно то, что прямого доступа к ним центральный процессор не имеет. Прежде чем воспользоваться информацией, хранящейся на ВЗУ, она должна быть перенесена в оперативную память. ВЗУ служат для хранения больших объемов информации, используются для создания библиотек, архивов, баз данных.

Каждый вышеперечисленный класс запоминающих устройств, в свою очередь, подразделяется на несколько типов, отличающихся своим функциональным назначением. В этом пункте рассмотрим несколько типов запоминающих устройств, входящих в состав **основной памяти** ЭВМ.

Состав основной памяти. В состав основной памяти ЭВМ входят:

1. **Оперативное** запоминающее устройство (ОЗУ), другое название - **запоминающее устройство с произвольной выборкой** (ЗУПВ), или в английской терминологии RAM (Random Access Memory). Оперативные запоминающие устройства всех классов ЭВМ представляют собой энергозависимые ЗУПВ.
2. **Постоянное** запоминающее устройство (ПЗУ). В английской терминологии такого рода память называют ROM (Read Only Memory) - память, работающая только на считывание. Информация, находящаяся в такой памяти, заранее закладывается при ее изготовлении ("зашивается") и при отключении питания не разрушается.

6.2.1. Оперативное запоминающее устройство

Структура ОЗУ. Общая схема структуры и функционирования ОЗУ, изображена на рис. 18.

Двоичный код адреса по шине адреса поступает на дешифратор адреса и возбуждает линию, идущую к выбранной ячейке памяти.

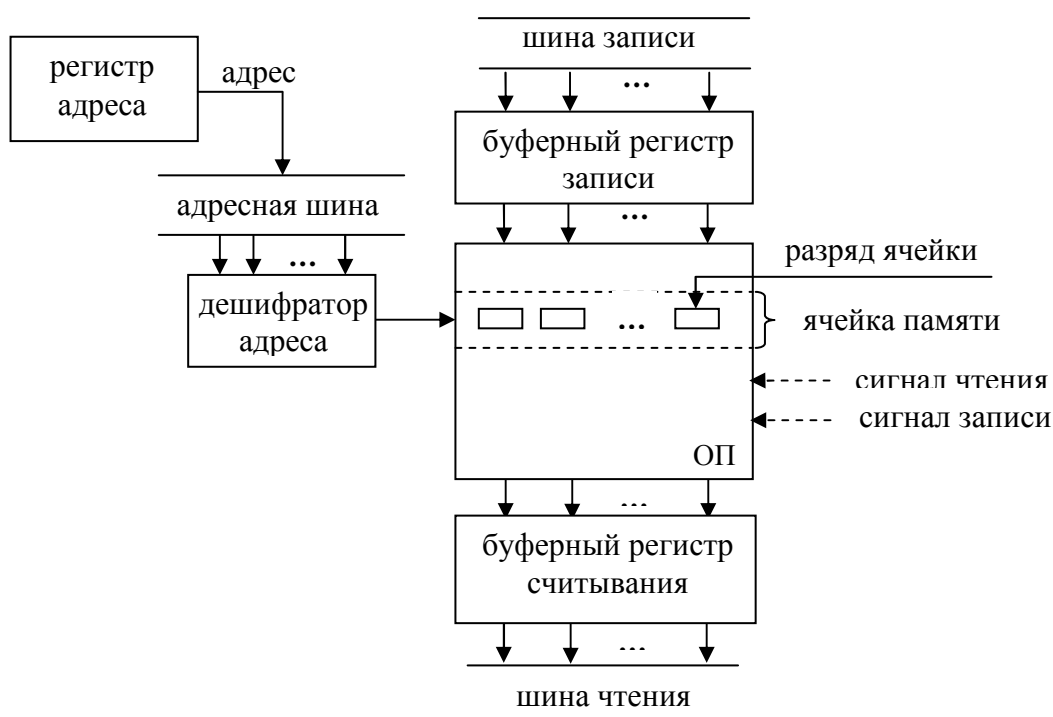
При операции чтения эта линия открывает вентили, обеспечивающие передачу значения, хранящегося в каждом разряде выбранной ячейки, в соответствующий разряд буферного регистра считывания.

При записи в ячейку памяти направление потока разрядной информации меняется на противоположное.

Режим считывания или записи задается управляющими сигналами, причем выполнение операций производится в соответствии с определенной последовательностью **тактов**. Например, для записи в ячейку с адресом А необходимо, чтобы:

- соответствующее данное было передано по шине записи в буферный регистр записи,
- адрес А поступил на дешифратор,
- сигнал управления записью был передан в ЗУ.

Очевидно, что для выполнения операции чтения первый этап опускается, так что в общем случае цикл чтения короче, чем цикл записи.



ОП – оперативная память.

Рис. 18. Упрощенная схема ОЗУ

Логическая структура оперативной памяти. Для пользователей (программистов) современная оперативная память ЭВМ представляет байтовую организацию, то есть адрес памяти является адресом (номером) ее соответствующего байта.

Адрес байта памяти	Байты памяти
0000	10011101
0001	01110010
0002	10101011
...	...

Рис. 19. Память из 8-ми разрядных ячеек (слов)

Очевидно, что размер байта, удобный для обработки символьной информации, крайне неэффективен при выполнении, например, арифметических операций. Поэтому в ОП можно логически выделить и более крупные хранилища (полуслово, слово, двойное слово), содержимое которых обрабатывают соответствующие машинные команды. Адресами полуслов, слов, и двойных слов является адрес левого байта, начиная с которого соответствующее хранилище располагается в памяти. Очевидно, что размер слова ОП равен размеру *машинного слова*. На рис. 20. определяется структура ОП с 32-х разрядным словом.

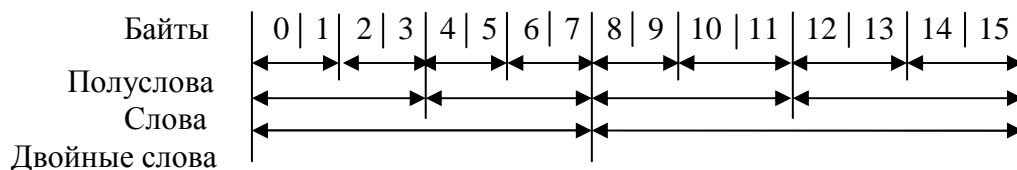


Рис.20. Логические хранилища информации в ОП

В дальнейшем условимся любое хранилище информации в общем случае называть ячейкой памяти, если не требуется специального уточнения ее размера.

6.2.2. Постоянное запоминающее устройство

Упрощенно можно считать, что общая структура ПЗУ отличается от структуры ОЗУ отсутствием регистра записи (рис. 21.).

Основное назначение ПЗУ - хранение системных констант и микропрограмм.

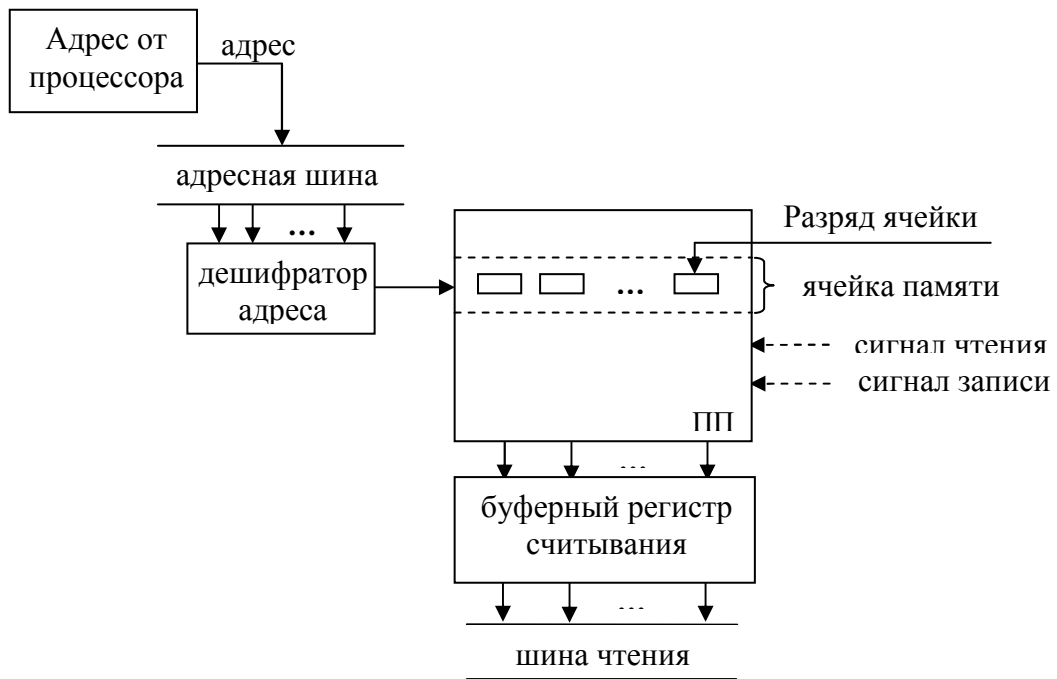
Классификация по методам доступа. Ранее отмечалось, что одной из характеристик памяти является способ размещения и поиска информации. Очевидно, что оперативная и постоянная памяти являются **адресными**, то есть доступ к ячейке памяти осуществляется по ее адресу. Рассмотрим запоминающие устройства, использующие методы доступа к памяти, основанные на других подходах.

6.2.3. Стековая память

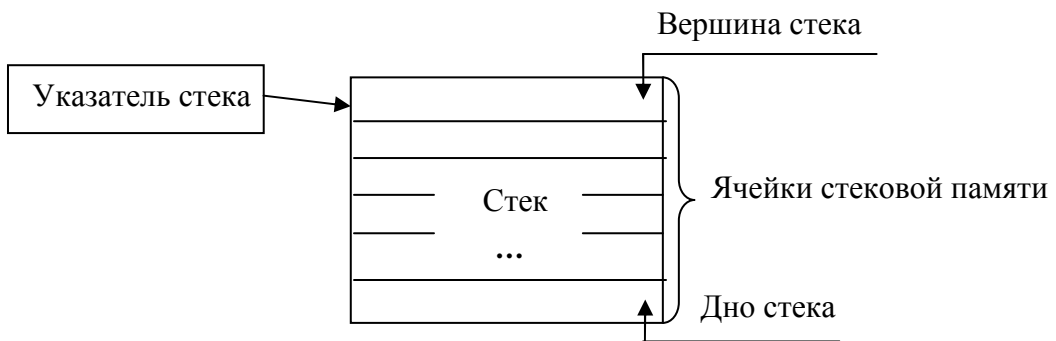
Структура стековой памяти (стека) и логика управления ею представлены на рис. 22.

Стековая память состоит из ячеек, причем обмен информацией между остальными устройствами ЭВМ и стеком всегда выполняется только через верхнюю ячейку – вершину стека. При записи нового слова (команды, числа, символа и т.п.) все ранее записанные слова сдвигаются на одну ячейку вниз, а новое слово помещается в вершину стека. Считывание возможно только с вершины стека и производится с удалением или без

удаления считываемого слова. Таковую память часто называют памятью типа LIFO (Last – In First – Out –последним вошел, первым вышел).



ПП – постоянная память, представляющая собой совокупность адресуемых ячеек.
Рис.21. Упрощенная структура ПЗУ



Указатель стека - регистр для указания адреса вершины стека.
Рис.22. Структура стека

Принципы функционирования стека проиллюстрированы на рис. 23.

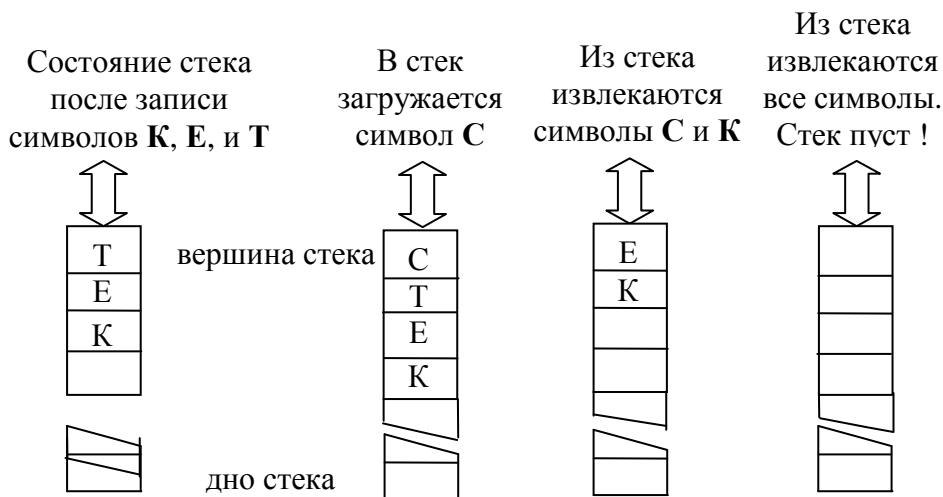


Рис.23. Пример работы со стеком

Очевидно, что аппаратная реализация стека, как правило, не целесообразна, поскольку при выполнении каждой операции содержимое стека перемещается целиком, поэтому в ЭВМ стек моделируется, и в качестве стековой памяти используется просто часть адресной памяти, что позволяет легко менять содержимое стека (рис. 24.).

Несмотря на то, что внешне модель стека отличается от аппаратного стека, она функционирует по тому же алгоритму: последним вошел - первым вышел. Это обеспечивается с помощью указателя стека, который содержит адрес плавающей вершины стека. Заметим, что моделируемый стек растет не сверху вниз (ко дну стека), а снизу вверх (от дна стека). Однако это не нарушает алгоритма функционирования такой памяти. На рис. 24. показан стек, организованный в памяти с 32-разрядными словами. Предполагается, что дно стека располагается в ячейке с 8-ричным адресом 1000.

При работе со стеком, представленным на рис. 24., следует учесть, что при занесении данного в стек сначала уменьшается значение указателя стека, то есть происходит его настройка на новую вершину, а затем уже производится загрузка в стек; при считывании из стека после извлечения данного из вершины указатель стека увеличивает свое значение. Именно поэтому, когда стек пуст, значение указателя устанавливается на ячейке, следующей за дном стека.

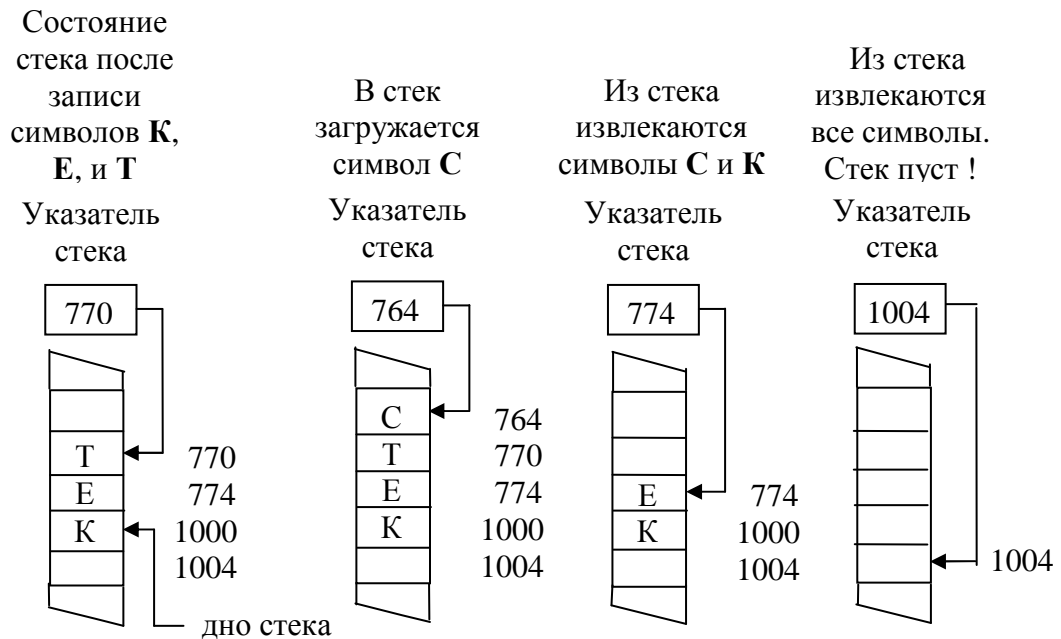


Рис. 24. Модель стека, реализованная в оперативной памяти

В качестве демонстрации приведем пример использования стека для вычисления арифметических выражений с использованием **Польской Инверсной Записи** (ПОЛИЗ). В ПОЛИЗе операция записывается не между операндами (X+Y), а после них (XY+).

Подобные выражения можно вычислять по следующему алгоритму:

- 1) проанализировать каждый символ бесскобочной записи формулы, начиная с крайнего левого символа, до тех пор, пока не встретится знак операции;
- 2) взять ближайших два операнда (если операция двуместная) или один операнд (при унарной операции), расположенные слева от обнаруженного знака операции, выполнить операцию и результат поместить в формулу на место выделенных операндов и знака операции;
- 3) если после выполнения пункта 2) формула состоит из одного значения, это значение и есть результат, то есть алгоритм завершен, в противном случае перейти к пункту 1).

Пример использования такого алгоритма для вычисления по формуле

$$Y = (A+B-C)/(E-D),$$

преобразованной в формулу

$$Y=ABC--+ED-/\quad (*)$$

приведен в нижеследующей таблице. Для определенности в ней выбраны следующие числовые значения операндов: A=8, B=2, C=5, E=6 и D=4.

Шаг	Формула, подлежащая расчету	Левый знак операции	Операнды	Результат	Новая формула после выполнения операции
1	8 2 5 - + 6 4 - /	-	2 и 5	10	8 10 + 6 4 - /
2	8 10 + 6 4 - /	+	8 и 10	18	18 6 4 - /
3	18 6 4 - /	-	6 и 4	2	18 2 /
4	18 2 /	/	18 и 2	9	9

На рис. 25. изображен расчет выражения (*) с использованием стека.

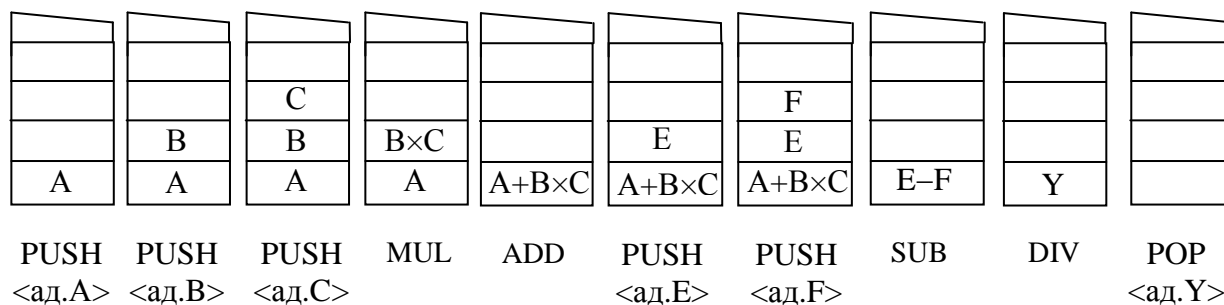


Рис. 25. Пример использования стека для вычисления выражений

Предполагается, что при работе со стеком на рис. 25. используются следующие команды:

- PUSH <ад.> - загрузка в стек содержимого ячейки с адресом <ад.>,
- POP <ад.> - выгрузка из стека в память по адресу <ад.>,
- MUL - умножение содержимого двух верхних ячеек стека,
- ADD - сложение содержимого двух верхних ячеек стека,
- SUB - вычитание содержимого двух верхних ячеек стека,
- DIV - деление содержимого двух верхних ячеек стека.

6.2.4. Ассоциативные запоминающие устройства

Ассоциативные запоминающие устройства (АЗУ) играют значительную роль в архитектуре современных ЭВМ для повышения производительности их работы. Описание назначения и сферы применения АЗУ требует специального рассмотрения, здесь ограничимся лишь изложением конструктивных особенностей данного типа памяти.

АЗУ обеспечивает возможность поиска информации, хранящейся в ассоциативной памяти (АП), по содержимому ячеек (ассоциативному признаку, контексту), а не по адресам. Упрощенная схема использования АЗУ представлена на рис. 26. (предполагаются 8-разрядные ячейки памяти).

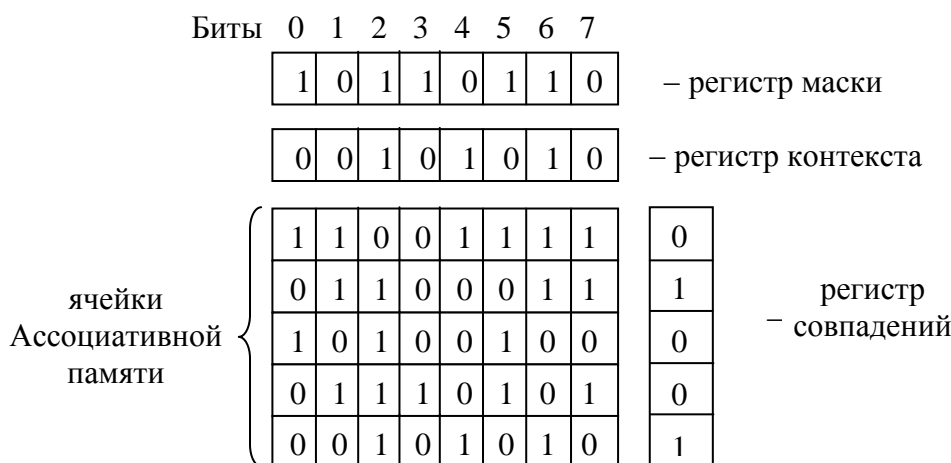


Рис. 26. Упрощенная структура АЗУ

Ассоциативная память состоит из сравнительно небольшого количества ячеек. В состав устройства управления такой памятью входят три специальных регистра:

- регистр маски,
- регистр контекста (или ассоциативного признака),
- регистр совпадений.

Размер регистров маски и контекста совпадает с размерностью ячеек ассоциативной памяти, а объем регистра совпадений равен количеству ячеек в ассоциативной памяти, причем каждый разряд регистра совпадений связан с соответствующей ячейкой ассоциативной памяти.

Назовем **маской** упорядоченную совокупность номеров разрядов регистра маски, содержимое которых установлено в 1. На рис. 26. маску образует совокупность номеров 0,2,3,5,6.

Назовем **контекстом** содержимое разрядов регистра контекста, номера которых совпадают с номерами разрядов маски. На рис. 26. контекст представлен следующей комбинацией битов: 0, 1,0,0, 1.

Будем говорить, что *i*-й разряд ячейки **замаскирован**, если его номер совпадает с номером одного из разрядов маски.

Ассоциативный поиск, осуществляемый в ячейках ассоциативной памяти, заключается в фиксации ячеек, для которых содержимое упорядоченной (по номеру разряда) совокупности замаскированных разрядов совпадает с контекстом. Фиксация выделенных ячеек осуществляется установкой в 1 содержимого соответствующих разрядов регистра совпадения.

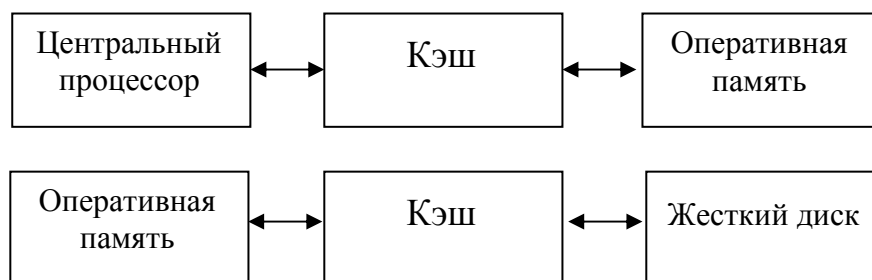
Важнейшим свойством АЗУ является то, что проверка содержимого проводится одновременно для всех ячеек ассоциативной памяти, что существенно ускоряет поиск по сравнению с последовательным адресным обращением к ячейкам ОЗУ. Следует, однако, отметить, что такое ускорение достигается за счет усложнения схемы управления памятью, поскольку для осуществления параллельного (одновременного) просмотра каждая ячейка должна снабжена локальным блоком управления, что резко повышает стоимость такой памяти. Именно поэтому объем ассоциативной памяти (по сравнению с объемом оперативной памяти) обычно невелик. В современных серийных моделях ЭВМ АЗУ нашли применение в некоторых архитектурах "сверхоперативных" запоминающих устройств, чаще называемых памятью типа Кэш.

6.2.5. Кэш-память

Кэш (англ. cache, от фр. cacher «прятать», т.е. «скрытая» память) — промежуточный буфер с быстрым доступом, содержащий информацию, которая может быть запрошена с наибольшей вероятностью. Предназначен Кэш для ускорения обращения к данным, содержащимся постоянно в памяти с меньшей скоростью доступа.

Таким образом, Кэш выступает в роли буфера, сглаживающего различие при взаимодействии в быстром действии высокоскоростного и медленного (в сравнении с высокоскоростным) устройств. Наличие такого буфера позволяет уменьшить время простоев (состояние ожидания) более быстрого устройства.

В настоящее время кэширование применяется очень широко, здесь упомянем лишь два примера взаимодействия устройств с использованием Кэш:



Структуру и принципы функционирования Кэш рассмотрим на примере основной взаимодействующей пары: ЦП – ОП.

Заметим, что в настоящее время соотношение скорости работы процессора и скорости работы оперативной памяти составляет приблизительно 1:1000, то есть, если данные, требуемые процессору для дальнейших вычислений, находятся в оперативной памяти (а не в Кэше), то процессор будет вынужден их ожидать, пропуская десятки и сотни операций. Если же данные находятся в Кэше, то они могут быть переданы процессору в ритме, необходимом для его безостановочной работы.

Состав Кэш и принципы функционирования представлены на рис.27.

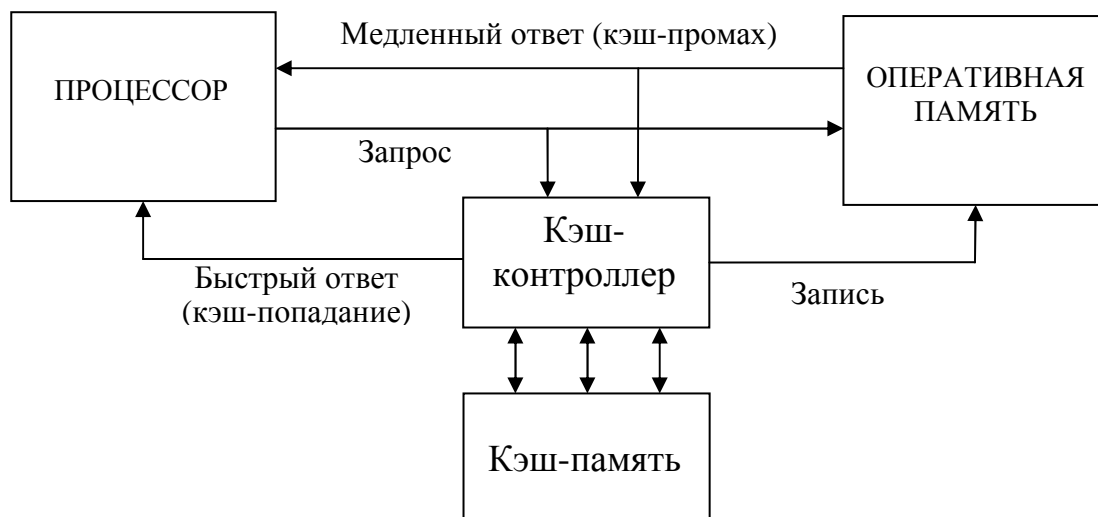


Рис.27. Принципы использования кэш-памяти

Кэш-контроллер управляет кэш-памятью: загружает в неё нужные данные из оперативной памяти и возвращает, когда нужно, модифицированные процессором данные в оперативную память.

Когда процессор хочет прочесть (или записать) данные по какому-либо адресу оперативной памяти, он передает этот адрес в контроллер Кэш-памяти. Контроллер по некоторому алгоритму определяет, содержатся ли в кэш-памяти данные, соответствующие полученному от процессора адресу. Если данные найдены (это событие называется попаданием в кэш), то Кэш-контроллер выдает требуемые данные процессору (в случае чтения), либо перезаписывает их полученными от процессора данными (в случае записи). Если же данные не найдены (промах Кэша), то производится обращение к оперативной памяти, и процессор вынужден ждать. Замечание. Важно понимать, что кэш всегда «полон», так как оставлять часть кэш-памяти «пустой» было бы совершенно нерационально. Поэтому перед запуском приложения Кэш заранее заполняется, а новые данные попадают в кэш только путём вытеснения (замещения) каких-либо старых данных.

Высокая степень попадания в кэш объясняется некоторыми объективными свойствами компьютерных данных:

- **Пространственная локальность.** Если произошло обращение по некоторому адресу, то с высокой степенью вероятности в ближайшее время произойдет обращение к соседним адресам. Пространственная локальность позволяет надеяться, что имеет смысл копировать целый блок данных в кэш, так как, скорее всего, в ближайшее время обращение будет к данным из этого блока.
- **Временная локальность.** Учет циклических участков программы. Если произошло обращение по некоторому адресу, то в ближайшее время с большой вероятностью произойдет обращение к этому же адресу.

В ячейку Кэша загружается содержимое целого блока (последовательности) ячеек ОП вместе с адресом этого блока. Учитывая последовательную выборку команд и

последовательную работу с элементами массивов данных, расположенных в ОП, после удачного попадания (размещения нужной ячейки в блоке, загруженном в Кэш), обращения к ОП не происходит до «исчерпания» блока.

Замечание. Очевидно, что последовательный процесс выполнения команд могут нарушить команды передачи управления. Если передача управления происходит на команду, расположенную в этом же блоке, проблем не возникает. В противном случае производится поиск требуемого нового блока, и при его отсутствии в Кэше фиксируется «промах», приводящий к необходимости его подкачки.

Алгоритм работы Кэша представлен на рис.28.

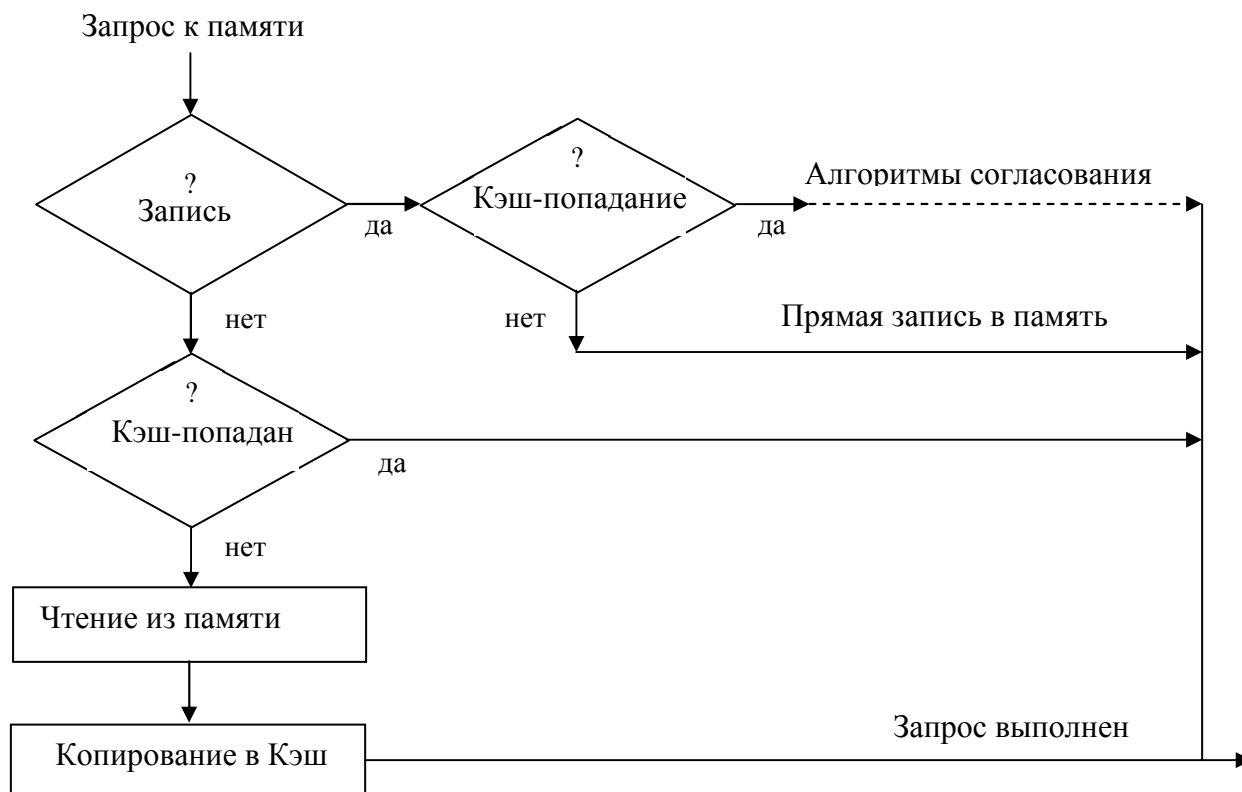


Рис.28. Алгоритм работы Кэш

Эффективность Кэш. Эффективностью Кэша называется отношение числа попаданий к общему количеству обращений процессора. Эффективность, таким образом, — это число от 0 до 1. Нулевая эффективность означает, что кэш несколько не ускорил работу системы; эффективность, равная единице, означает, что ускорение максимально, и время обращения к памяти определяется скоростью работы Кэша, а не скоростью работы оперативной памяти.

Замечание. В настоящее время вероятность попадания в кэш составляет не менее 90%.

Эффективность Кэша зависит от следующих факторов:

- 1) **Объём.** Чем больше объём Кэша, тем большую часть требуемых программе данных он может в себе содержать, тем реже будут происходить обращения к оперативной памяти.
- 2) **Алгоритм функционирования.** Зачастую объёма Кэш-памяти недостаточно для того, чтобы вместить все необходимые для вычислений данные. В этом случае Кэш-контроллер должен «решить», какие именно данные следует держать в Кэше. Поэтому, кроме объёма Кэша, важным является алгоритм его функционирования: кэш, оснащенный хорошим алгоритмом, будет гораздо эффективнее использовать свой объём, храня меньше ненужных данных.

- 3) **Выполняемая** процессором **программа**. Кэш оказывается эффективным потому, что большинство компьютерных программ обращаются к памяти не случайным образом, а закономерно. Чем лучше Кэш-контроллер может «предсказать» обращения приложения к памяти, тем выше эффективность. Закономерность обращений программы к оперативной памяти обычно связана с рассмотренными выше свойствами временной и пространственной локальности. Используя эти предположения, кэш-контроллер прогнозирует обращения процессора к памяти, заранее загружая в кэш-память необходимые данные.

Способы отображения оперативной памяти на Кэш-память. Содержание Кэш-памяти представляет собой совокупность записей обо всех блоках данных из основной памяти, загруженных в нее, включая их адреса (рис.29.).

Адрес данных в ОП	Блоки данных	Управляющая информация
-------------------	--------------	------------------------

Рис.29. Структура строки Кэш – памяти

Рассмотрим механизм отображения блоков ОП в Кэш-память.

Пример. Имеется ОП емкостью 256К байтов ($256К = 2^{18}$); для адресации такой памяти необходим 18-разрядный адрес.

Пусть ОП разбивается на блоки по 16 байтов в каждом. Очевидно, что ОП удобно рассматривать как линейную последовательность из $16384=2^{14}$ блоков.

При такой организации 18-разрядный адрес можно условно разделить на две части:

- младшие 4 разряда определяют адрес байта в пределах блока,
- старшие 14 разрядов определяют номер одного из 16384 блоков.

В дальнейшем старшие 14 разрядов адреса ОП будем называть **адресом блока ОП**.

Пусть имеется Кэш-память емкостью 2К байтов ($2К=2^{11}$). Таким образом, для адресации Кэш-памяти необходим 11-разрядный адрес.

В строку (ячейку) Кэш-памяти отображается блок ОП, следовательно, Кэш содержит $128=2^7$ слов.

11-разрядный адрес слова в Кэш-памяти можно представить состоящим из двух частей:

- адрес байта в строке (4 младших разряда),
- адрес строки Кэш-памяти (7 старших разрядов).

Так как расположение байтов в блоке ОП и строке Кэш одинаково, доступ к конкретному байту в строке Кэш-памяти определяется 4 младшими разрядами адреса ОП.

Остается задача преобразования 14-разрядного адреса блока ОП в 7-разрядный адрес строки Кэш-памяти, то есть способа определения взаимного соответствия строки Кэш-памяти и области ОП. В этой связи различают три архитектуры Кэш-памяти:

- Кэш прямого отображения,
- полностью ассоциативный Кэш,
- комбинация первых двух подходов - частично-ассоциативный Кэш.

Для простоты ограничимся рассмотрением только архитектуру Кэш прямого отображения.

Прямое отображение. При прямом отображении адрес *i*-й строки Кэш-памяти, на которую может быть отображен *j*-й блок ОП, однозначно определяется выражением $i = j \bmod m$, где *m* – общее количество строк в Кэш-памяти. В примере $i = j \bmod 128$, где *i* может принимать значения от 0 до 127, а *j* – от 0 до 16383.

Таким образом, на строку Кэш-памяти с номером *i* отображается каждый из 128 блоков ОП, начиная с номера *i*. (рис.30.).

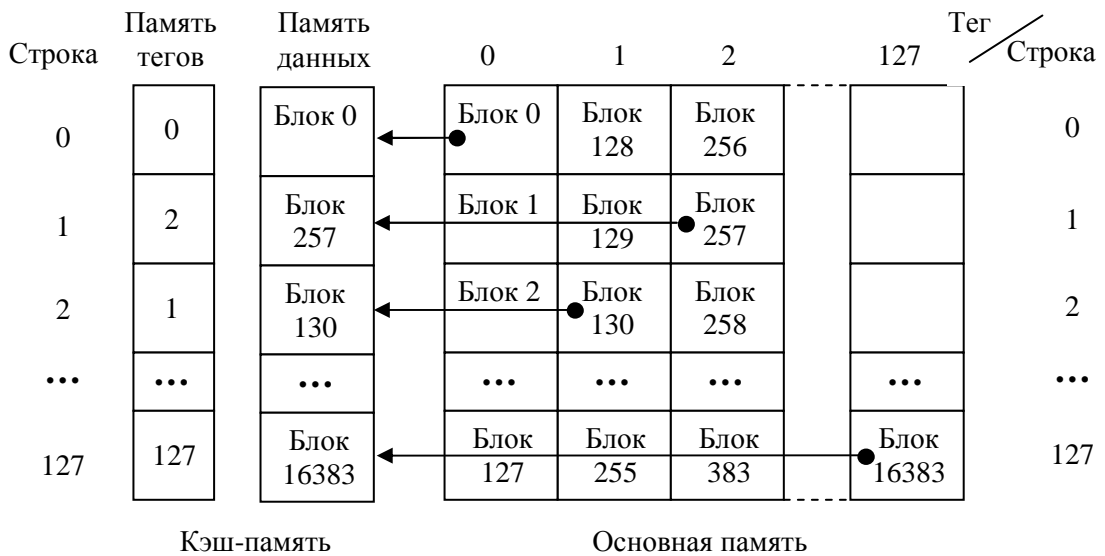


Рис.30. Организация кэш-памяти с прямым отображением

На рисунке ОП условно представлена в виде двухмерного массива блоков, в котором количество рядов равно количеству строк в Кэш-памяти, а в каждом ряду последовательно перечислены блоки, загружаемые на одну и ту же строку Кэш-памяти.

При реализации такого отображения логика Кэш-памяти условно разбивает 14-разрядный адрес блока ОП на два поля:

- 7-разрядное поле строки,
- 7-разрядный тег.

Поле строки указывает на одну из $128=2^7$ строк Кэш-памяти, в которую может быть отображен блок с заданным адресом.

Поле тега определяет, какой именно из списка блоков, закрепленных за данной строкой Кэш, будет отображен.

Более точно, строка кэш-памяти прямого отображения может быть устроена следующим образом (рис. 31.).

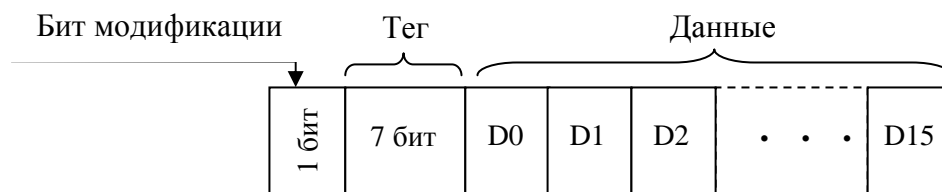


Рис.31. Строка Кэша с прямым отображением адресов

В общем виде при прямом отображении работа кэш-контроллера представляется следующей последовательностью действий.

Выполнение запроса процессора **на чтение** байта оперативной памяти:

1. разбивка адреса требуемого байта на части для определения тэга, номера строки Кэша и смещения в строке;
2. сверка тэга, записанного в нужной строке Кэша, с тэгом, выделенным из адреса;
3. если тэги не совпали (данные в Кэше отсутствуют), то выполняются следующие действия:
 - а) если бит модификации строки не равен нулю, то он обнуляется, и содержимое строки сохраняется - записывается в оперативную память по адресу, определяемому хранящимся в строке тэгом и номером этой строки в Кэше;

- b) в строку Кэша из оперативной памяти загружаются 16 байт, среди которых находится требуемый процессором байт памяти (адрес этой последовательности байт равен адресу требуемого байта, но с обнуленными последними четырьмя битами);
 - c) тэг строки заменяется на тэг, выделенный из адреса требуемого байта;
4. теперь требуемые данные точно присутствуют в Кэше, и контроллер выдает байт процессору из строки Кэша в соответствии со смещением, выделенным из адреса.

Выполнение запроса процессора **на запись** (модификацию) байта в оперативную память:

1. разбивка адреса требуемого байта на части для определения тэга, номера строки Кэша, и смещения в строке;
2. сверка тэга, записанного в нужной строке Кэша, с тэгом, выделенным из адреса;
3. если тэги не совпали (перезаписываемые данные не прокэшированы), то выполняются следующие действия:
 - a) если бит модификации строки не равен нулю, то содержимое строки записывается на место оперативной памяти, определяемое хранящимся в строке тэгом и номером этой строки в Кэше;
 - b) в строку Кэша из оперативной памяти загружаются 16 байт, среди которых находится перезаписываемый процессором байт памяти (адрес этой последовательности байт равен адресу требуемого байта, но с обнуленными последними четырьмя битами);
 - c) тэг строки заменяется на тэг, выделенный из адреса требуемого байта;
4. теперь перезаписываемые данные точно присутствуют в Кэше, и контроллер модифицирует байт строки Кэша в соответствии со смещением, выделенным из адреса;
5. бит модификации строки устанавливается в 1.

Кэш с прямым отображением адресов, будучи самым простым с точки зрения аппаратной реализации, исторически был первым распространенным видом Кэшей.

Достоинства подхода – простота и дешевизна реализации.

Недостаток – жесткое закрепление за различными блоками ОП одной строки Кэш-памяти. Поэтому, если программа поочередно обращается к словам из двух разных блоков, отображающихся на одну и ту же строку Кэш-памяти, вероятность частого обновления Кэш будет очень большой.

7. ПОДСИСТЕМА УПРАВЛЕНИЯ

7.1. Основные функции и состав устройства управления

Упрощенная структура процессора и схема выполнения машинной команды обсуждены в п.5.2. В этом разделе рассмотрим подробнее структуру и принципы функционирования устройства управления (УУ).

В общем случае УУ формирует управляющие сигналы для выполнения следующих функций:

- выборка из основной памяти (ОЗУ или ПЗУ) кода очередной команды;
- расшифровка кода операции и признаков выбранной команды;
- формирование исполнительного адреса операнда;
- выборка операндов и выполнение машинной операции;
- анализ запросов на прерывание исполняемой программы;
- формирование адреса следующей команды.

Для выполнения указанных функций УУ содержит блоки: расшифровки команд, адреса команды, управления операциями, прерываний, тактовых импульсов, на входы которых поступают определенные группы управляющих сигналов (УС) (рис. 32.).

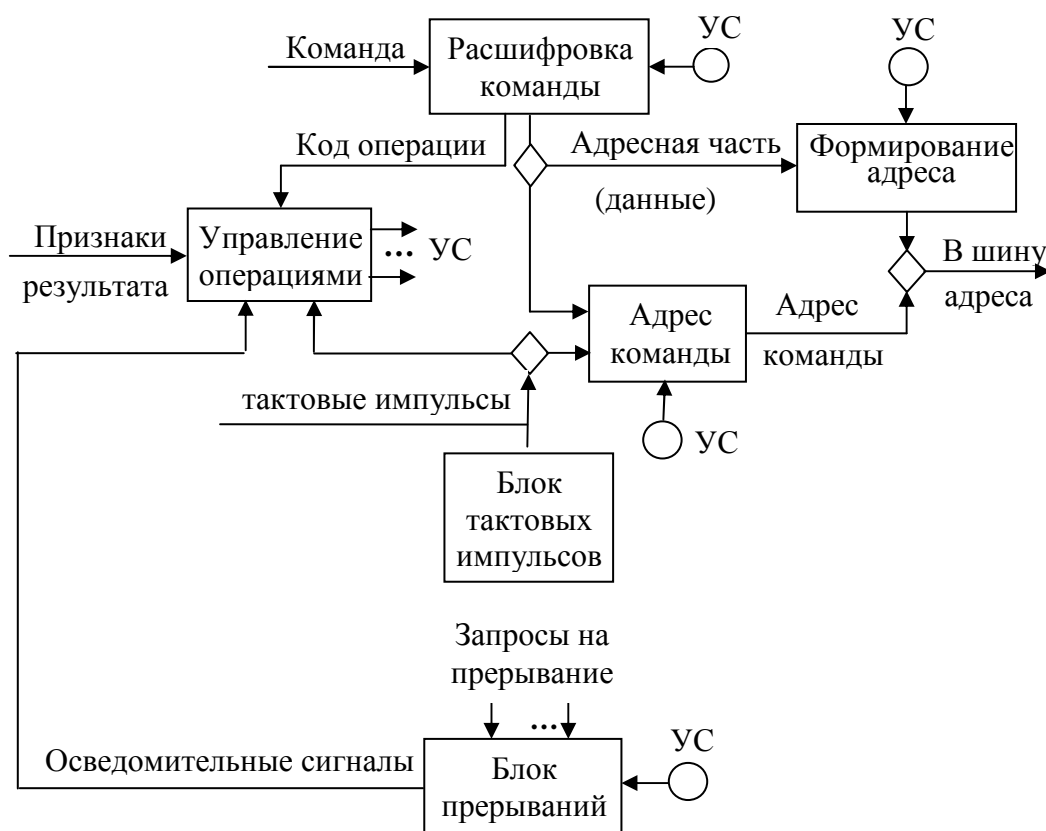


Рис. 32. Общая структура и функционирование устройства управления

По соответствующим управляющим сигналам адрес очередной команды из блока адреса команды (регистр счетчика команд) пересылается в ОЗУ (или ПЗУ), устанавливаемое в режим считывания. По этому адресу очередная команда программы принимается для расшифровки и исполнения в блок расшифровки команд (регистр команд). Управляющие сигналы, под воздействием которых производится выдача и прием адреса команды и собственно кода команды (ее двоичного представления), непосредственно вырабатываются блоком тактовых импульсов и называются **тактовыми импульсами**. Блок тактовых импульсов вырабатывает и сигналы общего управления режимами работы всех блоков процессора и запоминающих устройств.

После приема кода команды в блок регистра команд ее операционная часть (код операции, признаки) пересылаются в блок управления операциями, который после расшифровки операционной части вырабатывает последовательность управляющих сигналов; эти управляющие сигналы инициируют операции арифметико-логического устройства процессора и синхронизированные тактовые импульсы, необходимые для выполнения требуемой машинной операции.

Если операндная часть кода команды содержит ссылки на адрес основной памяти, блок формирования адреса обеспечивает формирование исполнительных адресов операндов. Следует учесть, что в командах перехода операндная часть указывает на адрес следующей команды программы, в этом случае адрес поступает на вход блока адреса команды.

В современных ЭВМ перед выборкой очередной команды, а также во время ее выполнения производится анализ запросов на прерывание текущей программы (в частности, на наличие поступивших сигналов от внешних запоминающих устройств на

выполнение операций ввода-вывода), для чего в устройство управления включается **блок прерываний**.

7.2. Принципы функционирования

Выполнение машинных команд. В процессе **исполнения** машинных команд устройство управления производит анализ и пересылку команды, отдельных ее частей (кода операции, признака способа адресации и адреса) или операнда из одного регистра в другой, арифметико-логическое устройство, память. Эти элементарные действия (микрооперации) протекают в определенной временной последовательности и скоординированы между собой.

Микрооперации осуществляют:

- передачу информации из одного регистра в другой,
- выполнение элементарных сдвигов в рамках одного регистра,
- проверку бита в регистре,
- передачу данных из регистра в шины и т.д.

Каждая машинная команда выполняется в течение некоторого временного отрезка (**цикла команды**), определяемого количеством тактовых импульсов работы ЭВМ, необходимых для ее (команды) реализации. Цикл команды включает один или несколько **машинных циклов**, причем каждый машинный цикл предназначен для определенной цели (выполнения функции УУ), например, **выборка команды, дешифровка кода операции, выборка операнда** и т.д. Ясно, что каждый машинный цикл требует выполнения определенной последовательности микроопераций.

На вход микрооперации поступает булев вектор, представляющий значение соответствующей компоненты команды (код операции, номер регистра, адрес памяти). Результатом выполнения микрооперации является булев вектор, задающий набор сигналов, которые должны быть поданы на вход соответствующих управляющих линий.

Таким образом, самый высокий уровень иерархии управления состоит в анализе информации, содержащейся в машинной команде; саму машинную команду, в конечном счете, представляющую собой булев вектор, в некотором смысле можно рассматривать как совокупность управляющих сигналов. Выполнение микрооперации в этом случае есть не что иное, как результат реализации соответствующей комбинационной схемы.

Основы функционирования операционной компоненты. При разделении процессора на **управляющую компоненту** - устройство управления (УУ), и **операционную компоненту** (блок управления операциями – БУО или АЛУ), его можно представить в следующем виде (рис. 33.)

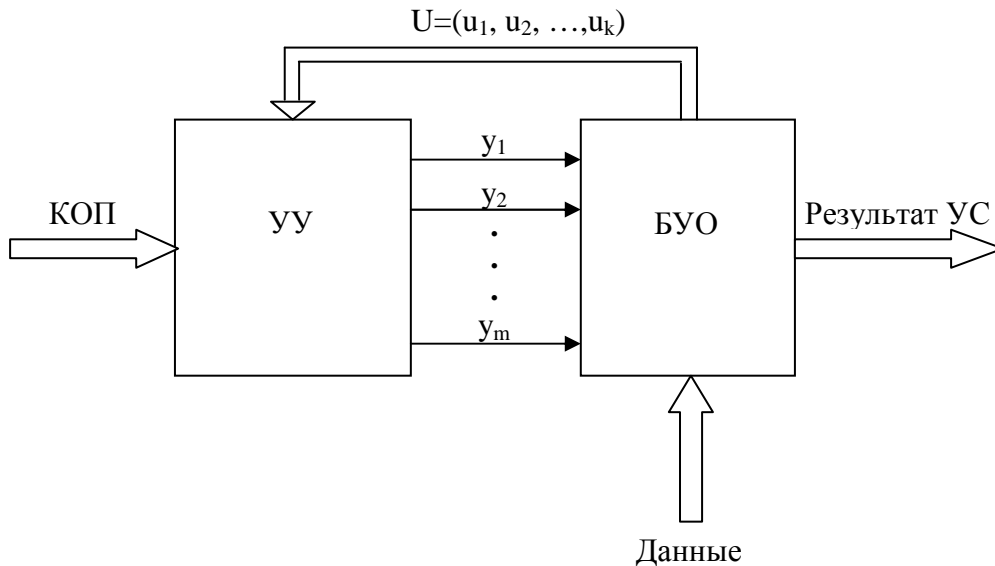


Рис. 33. Структура процессора при декомпозиции его на УА и ОУ

На приведенном рисунке $Y = \{y_1, y_2, \dots, y_m\}$ - множество функциональных сигналов, управляющих ходом выполнения операций. Каждый функциональный сигнал соответствует выполняемой микрооперации на некотором такте работы ЭВМ. $U = \{u_1, u_2, \dots, u_k\}$ - сигналы, оповещающие о ходе выполнения операции.

Функционирование блока управления операциями определяется следующими двумя множествами:

- множеством наборов сигналов на входах x_1, x_2, \dots, x_n , соответствующих двоичному коду операционной части команды (например, КОП) и двоичным значениям сигналов осведомительных признаков (на рис.33. вектор U);
- множеством двоичных управляющих сигналов на выходах $УС_1, УС_2, \dots, УС_m$ (на рис. 33. вектор Y), соответствующих множеству выдаваемых из устройства управления сигналов микроопераций: их общее число определяется количеством управляемых точек, в которые устройство управления выдает управляющие сигналы; эти выходные сигналы могут быть поданы на вход устройства управления более низкого уровня иерархии или непосредственно на вход управляемого блока.

Поскольку каждый из входных и выходных сигналов может принимать только два значения: 0 или 1, каждый выходной сигнал $УС_i$ можно описать как логическую функцию входных сигналов:

$$УС_i = f_i(x_1, x_2, \dots, x_n), \quad i=1, 2, \dots, m \quad (*)$$

По принципу формирования и развертывания временной последовательности управляющих сигналов (способу построения цикла работы ЭВМ и ее устройства управления), различают устройства управления **аппаратного** и **микропрограммного** типов.

В устройстве управления **аппаратного (схемного)** типа блок управления операциями представляет собой комбинационную схему, в которой требуемое множество состояний задается множеством логических и запоминающих элементов, непосредственно (без вмешательства микропрограмм), реализующих функции переходов и выходов. Другими словами, логические функции (*) реализуются электронными схемами из элементов, выполняющих простейшие логические действия **И**, **ИЛИ**, **НЕ**. Для каждого такта обеспечивается выбор требуемых элементов, что позволяет подать на выходы блока сигналы микроопераций. Ясно, что всякое изменение управления требует изготовления новой комбинационной схемы и (что эквивалентно) перепроектирования ЭВМ.

В устройстве управления *микропрограммного типа* логические функции (*) рассматриваются как некоторые *формулы*.

Например, комбинационную схему полусумматора можно представить следующей совокупностью формул:

$$S = \overline{X \wedge Y} \wedge (X \vee Y)$$

$$C = X \wedge Y$$

Очевидно, что значение управляющих сигналов S и C можно "вычислить", используя блок управления операциями как электронную машину с упрощенной системой команд (микрокоманд), выполняющей простейшие логические операции (микрооперации); микрооперации выполняются в соответствии с указаниями, содержащимися в микрокомандах. Таким образом, блок управления операциями выполняет функции блока хранения и выборки кодов микрокоманд.

Микропрограммный принцип управления представляет более гибкие возможности для перенастройки центрального процессора при частичной модификации системы команд; для этого достаточно поменять микросхему с ПЗУ, в которой "защита" система микропрограмм на эту же микросхему, но с другой настройкой ("прошивкой"). Такие изменения при схемной реализации невозможны. Однако схемная реализация существенно превосходит микропрограммную по быстродействию. В современных ЭВМ общего назначения (универсальных ЭВМ) используется микропрограммное управление, а в специализированных ЭВМ для достижения максимальной производительности - схемное.

Замечание. Совершенствование технологической базы в настоящее время позволяет упростить схемную перенастройку, что определяет тенденцию к постепенному переходу к аппаратному принципу управления (возврату к принципам управления в первых поколениях ЭВМ).

7.3. Понятие микропрограммирования

Микропрограммный принцип реализации устройства управления основан на том, что любую комбинационную схему можно реализовать запоминающим устройством. Как всякая комбинационная схема, так и всякая система логических функций могут быть однозначно заданы таблицей истинности.

x_0	x_1	...	x_{n-1}	f_0	f_1	...	f_{m-1}
0	0		0	0	1		1
0	0		1	1	1		0
...
1	1		0	0	0		0
1	1		1	0	1		1

Эта таблица задает соответствие между наборами значений n входных переменных x_0, x_1, \dots, x_{n-1} и наборами, характеризующими значения выходных функций f_0, f_1, \dots, f_{m-1} .

Рассмотрим структуру запоминающего устройства (ЗУ) на рис.34.

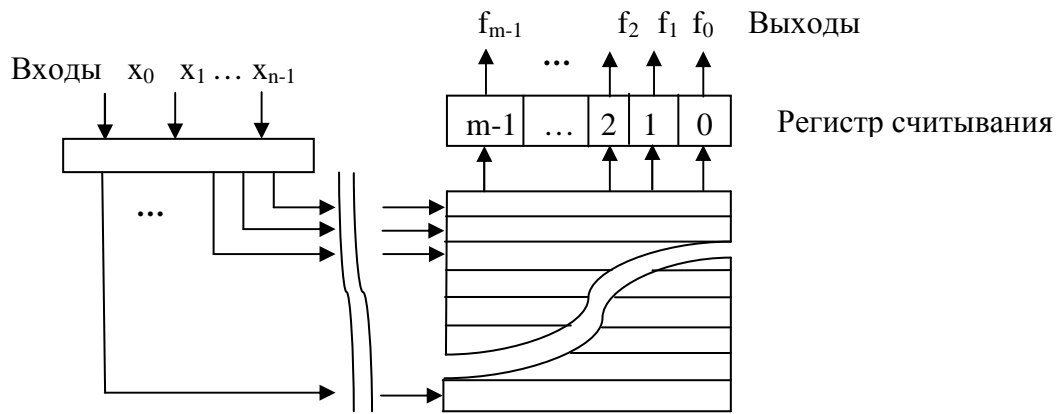


Рис. 34. Схема запоминающего устройства, реализующего комбинационную схему

На вход устройства поступает булев вектор $x=(x_0, x_1, \dots, x_{n-1})$, задающий номер (адрес) ячейки памяти.

Память состоит из 2^n m -разрядных ячеек, каждая из которых хранит вектор $f=(f_0, f_1, \dots, f_{m-1})$.

Такое ЗУ по своим функциям может заменить любую комбинационную схему с n входами и m выходами. Действительно, последовательность управляющих сигналов (содержимое ячеек), поступающих из памяти такого ЗУ, можно рассматривать как некоторую программу, определяющую функционирование управляющих схем дискретных устройств. Набор управляющих сигналов f несет информацию о тех микрооперациях, которые должны быть выполнены устройствами в данный временной такт. Справедливо поэтому назвать строки памяти микрокомандами, а их объединение - микропрограммой.

Каждый управляющий сигнал (*) связан с конкретной микрооперацией, которая должна быть выполнена, если он принял значение 1. Следовательно, двоичный набор, представляющий микрокоманду, можно рассматривать как перечень тех микроопераций, которые возбуждаются устройством управления для параллельного, одновременного выполнения в различных устройствах ЭВМ в данный такт работы.

Ячейки памяти часто называют *управляющими словами*, а саму память - *управляющей памятью*.

Замечание. В составе микрокоманды есть поле, хранящее адрес микрокоманды, которая должна выполняться на следующем такте соответствующего машинного цикла. С помощью этого поля микрокоманды и объединяются в микропрограммы.

В некоторых различных по своему функциональному назначению циклах работы ЭВМ выполняются группы одинаковых микрокоманд. Например, при разных по смыслу обращениях в основную память за командой или за данными необходимо выполнить одинаковую группу микрокоманд, формирующих адрес по памяти, вызывающих этот адрес на дешифратор оперативной памяти с учетом сигналов ее готовности к работе. Такие группы микрокоманд формируются в микроподпрограммы. Возможность использовать подпрограммы значительно экономит микропрограммную память. В то же время это требует усложнения микропрограммного устройства управления. В нем в том или ином виде должны существовать аппаратные средства, обеспечивающие переход к подпрограммам, возврат из подпрограмм, команды условного и безусловного переходов.

8. УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

8.1. Системная магистраль "Общая шина". Контроллеры внешних устройств и уровни сопряжения

Связь процессора с внешними устройствами осуществляется по шинам - системам проводов, несущим сигналы от процессора к внешним устройствам и обратно. Организация

связи в различных моделях ЭВМ различна; здесь описывается способ связи, принятый в наиболее распространенном классе современных ЭВМ – микро-ЭВМ.

Замечание. Микро-ЭВМ – это компьютеры, в которых центральный процессор выполнен в виде одной микросхемы (микропроцессора).

Линии, по которым происходит передача информации, можно разделить на три группы: **информационную** (шину данных), **адресную** (шину адреса) и **управляющую** (шину управления). В целом все эти шины носят название **Системная шина (Общая магистраль)**. При магистральном принципе организации ОЗУ и все внешние устройства подключаются к общей магистрали, исходящей из процессора (рис.35.).

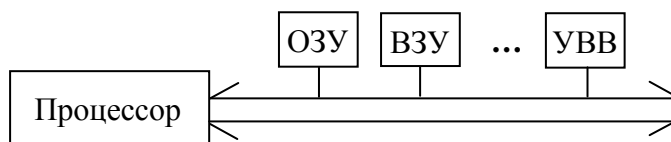


Рис.35. Магистральная организация архитектуры ЭВМ

Описание связей, осуществляемых по общей шине, должно включать ответы на ряд чрезвычайно важных вопросов:

- Каков способ передачи данных (словами, байтами, битами)?
- Каков способ передачи адреса?
- Какими управляющими и (или) осведомительными сигналами сопровождается передача адреса и данных?
- Какие осведомительные сигналы (о состоянии памяти и внешних устройств) пересылаются в процессор?
- Как должны быть распределены во времени информационные, управляющие и осведомительные сигналы?
- Какими напряжениями кодируются двоичные 0 и 1?
- С каких выводов системной платы или разъемов можно снять тот или иной сигнал и на какие подать?
- Какой тип разъема надо использовать для подключения к процессору?

Ответы на эти и многие другие вопросы можно получить, ознакомившись с **системным интерфейсом**.

В общем случае под **интерфейсом** понимается совокупность средств, определяющих логический порядок взаимодействия систем (протокол взаимодействия, в котором все физические и логические параметры согласуются между собой), и вытекающие из протокола требования к аппаратуре и к программному обеспечению, если обмен данными осуществляется под управлением программы.

Замечание. Если интерфейс является общепринятым, например, утверждённым на уровне международных соглашений, то он называется **стандартным**.

Таким образом, системный интерфейс - это набор цепей, связывающих процессор с памятью и внешними устройствами, алгоритм передачи сигналов по этим цепям, их электрические параметры и тип соединительных элементов.

Подключение любого внешнего устройства к ЭВМ осуществляется через его устройство управления - **контроллер ВУ** (далее просто контроллер). Таким образом, в системах ввода-вывода большинства современных ЭВМ можно выделить два уровня сопряжения внешних устройств с процессором и памятью. На первом уровне контроллеры сопрягаются с процессором и памятью через системный интерфейс, который обеспечивает объединение отдельных устройств ЭВМ в единую систему. На втором уровне сопряжения контроллеры посредством шин связи соединяются с соответствующими внешними устройствами (рис. 36.).

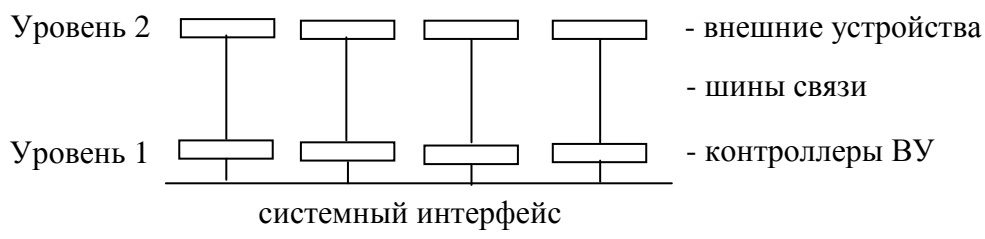


Рис.36. Уровни сопряжения контроллеров ВУ

Двухуровневая система интерфейса определяет зависимость способов структурной и функциональной организации контроллеров от двух основных факторов:

- форматов данных и режима работы конкретных внешних устройств,
- типа системного интерфейса (организации общей шины).

Влияние первого фактора на организацию контроллеров достаточно очевидна. Действительно, нерационально было бы создавать единый универсальный контроллер, обеспечивающий, например, подключение к ЭВМ простых устройств типа клавиатуры и сложных устройств типа накопителей на магнитных дисках. Именно поэтому на практике применяют самые разнообразные контроллеры: от простейших для подключения к ЭВМ датчиков одиночных сигналов до очень сложных, сравнимых по сложности с процессорами микро-ЭВМ, например, для сопряжения ЭВМ с аппаратурой для автоматизации научных исследований.

Второй фактор - тип интерфейса - определяет способ организации электронных схем контроллеров, обеспечивающих связь с шинами общей магистрали.

Упрощения и унификации аппаратуры сопряжения добиваются за счет введения промежуточного стандартного интерфейса параллельной или последовательной передачи данных. В этом случае для подключения к микро-ЭВМ различных внешних устройств с такими стандартными интерфейсами можно использовать одинаковые (для данной ЭВМ) контроллеры параллельного или последовательного интерфейса. Такие контроллеры, выпускаемые серийно в виде отдельных микросхем, часто называют **адаптерами**.

Так как у терминов интерфейс, контроллер, адаптер нет общепринятых четких определений, они часто считаются синонимами.

Основу контроллера составляют несколько регистров, которые служат для временного хранения передаваемой информации (рис.37.). Регистры входных и выходных данных работают соответственно только в режиме чтения (или ввода - перемещения информации из контроллера на системные шины) и только в режиме записи (или вывода - перемещения информации с системной шины в контроллер). Регистр состояния работает только в режиме чтения и содержит информацию о текущем состоянии внешнего устройства (включено/ выключено, готово/не готово к обмену данными и т.п.). Регистр управления работает только в режиме записи и служит для приема из ЭВМ приказов для внешнего устройства.

Сигналы Ввод и Вывод поступают в контроллер от центрального процессора и служат запросом на выполнение контроллером соответствующих операций с внешним устройством. Сигналом Готовность ВУ контроллер сигнализирует процессор об окончании выполнения текущей операции.

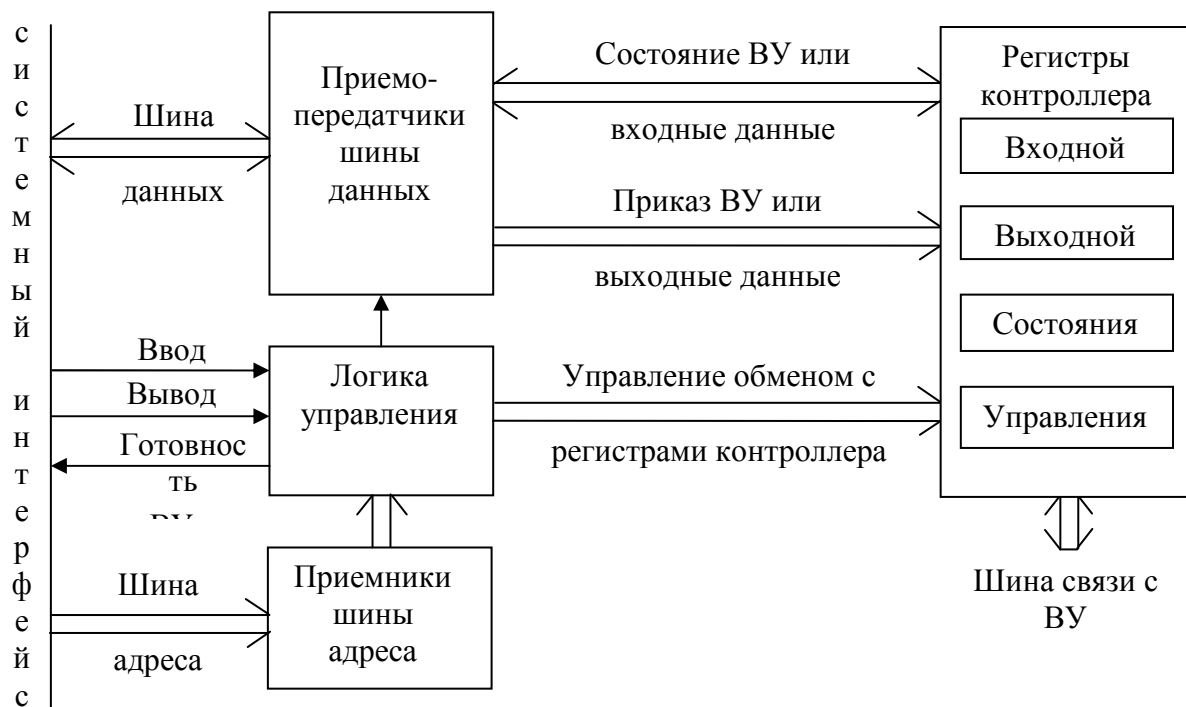


Рис. 37. Блок-схема контроллера ВУ для системного интерфейса

8.2. Управление магистралью "Общая шина"

Общая организация. Управление магистралью во многих современных универсальных ЭВМ осуществляется по аналогии с обращением к памяти. Ниже рассматривается именно эта схема.

Для адресов внешних устройств отведена часть единого адресного пространства системной магистрали, то есть, для каждого регистра каждого контроллера выделен уникальный адрес (аналогичный адресам основной памяти). Таким образом, в контроллерах удастся легко организовать селекцию (выбор) адресов собственных регистров из всего множества адресов, передаваемых по линиям адресной шины.

Запрос во внешнее устройство на ввод или вывод информации поступает по управляющим шинам **Чтение** и **Запись**, обеспечивающим обмен информацией процессора с модулями памяти. Кроме указания типа операции (ввод или вывод) запрос включает адрес **порта**.

При выполнении операции **вывода** (записи на устройство) адресуемый контроллер принимает данные, переданные ему процессором, пересылает их во внешнее устройство и соответствующим сигналом по шине управления (назовем его **Готовность ВУ**) сообщает процессору, что данные приняты внешним устройством и можно снять информацию с шин адреса и данных.

При выполнении операции **ввода** (чтения с устройства) адресуемый контроллер считывает порцию данных из носителя, выставляет на линиях шины данных значение разрядов считанного слова и сигналом **Готовность ВУ** сообщает об этом процессору. Приняв данные из контроллера, процессор снимает сигналы с шины адреса и данных (рис. 38.).

Таким образом, управляющий осведомительный сигнал **Готовность ВУ** может служить для синхронизации работы процессора и контроллеров внешних устройств, точнее, для указания моментов времени, определяющих готовность данных во внешнем устройстве для передачи либо подтверждающих их прием.



Рис. 38. Простейшее сопряжение контроллера с системным интерфейсом

Передача данных между любыми двумя устройствами, подключенными к магистрали Общая шина, осуществляется по принципу "управляющий - управляемый" ("задатчик - исполнитель", "ведущий - ведомый").

Как правило, хозяином магистрали является сам процессор. Перехватить управление магистралью разрешается лишь устройствам прямого доступа к памяти (например, контроллеру винчестерского диска) и лишь на период обмена с ОЗУ большими порциями данных (внепроцессорная передача данных).

Порты ввода-вывода. Адресное пространство ввода-вывода организовано в виде портов. Если в передаче информации участвует процессор, то направление потока входной или выходной информации принято рассматривать относительно самого процессора.

Входной порт - это любой источник данных (например, регистр контроллера), который подключается к Общей шине и из которого посылается слово данных в процессор.

Выходной порт - приемник данных (например, регистр контроллера), который подключается к Общей шине и, будучи выбранным, принимает слово данных из процессора.

Для согласования интерфейсов периферийные устройства подключаются к шине через свои контроллеры (адаптеры) и порты примерно по такой схеме:



Циклы магистрали. Обмен данными по Общей шине осуществляется с помощью одного или нескольких **циклов магистрали**, за каждый из которых по шинам адреса данных передается одно машинное слово (между ОЗУ и процессором) или байт данных (между ОЗУ и внешним устройством).

Каждый цикл Общей шины имеет свой **протокол**, то есть порядок обмена сигналами между ведущим и ведомым устройствами. Различными видами (типами) циклов являются **Ввод данных**, **Вывод байта**, **Вывод слова**, **Ввод-пауза-вывод**. Общим для всех видов циклов является их **асинхронный** характер. Это означает, что на каждый управляющий сигнал задатчика должен поступить ответ от исполнительного устройства (обмен

"рукопожатием") с фиксированным интервалом времени ответа (**таймаут**). Если в течение таймаута задатчик не получил ответа, происходит [внутреннее прерывание](#).

Любой из циклов магистрали начинается с адресации ячейки ОЗУ или регистра контроллера, к которым будет происходить обращение.

В ходе **адресной части** цикла процессор устанавливает нужный адрес на шине адреса и вырабатывает сигнал синхронизации адреса, который действует в течение всего цикла. С появлением сигнала синхронизации переданный адрес воспринимается всеми управляемыми устройствами, подключенными к Общей шине, и адресная часть цикла заканчивается.

В ходе последующей **информационной части** цикла выполняется конкретная операция ввода, вывода и т. п. В информационной части цикла принимает участие только тот исполнитель, который имеет в своем составе ячейку (или регистр) с указанным в адресной части цикла адресом.

8.3. Программно-управляемая передача информации

Устройства ввода-вывода передают в ЭВМ и получают из нее, как правило, большой объем информации, который не может быть размещен только в регистрах процессора. Поэтому информация передается из внешнего устройства в оперативную память и поступает на внешнее устройство из оперативной памяти. При этом обмен может идти под управлением программы (**программно-управляемая передача данных**) или под управлением специального внешнего устройства (**контроллера прямого доступа к памяти**), минуя процессор ([внепроцессорный обмен](#)).

При использовании программно-управляемого обмена должна быть составлена программа для пересылки данных из памяти ЭВМ в регистр данных контроллера или в обратном направлении.

В этой программе можно реализовать один из трех видов подобного обмена: **синхронный**, **асинхронный** и обмен **по прерыванию**.

Синхронный обмен можно использовать лишь для связи с такими внешними устройствами, для которых точно известно время выполнения операции (например, максимально возможное время, затрачиваемое на печать одного символа). При этом программу обмена необходимо составлять так, чтобы команды на обмен шли с интервалами не меньшими, чем время выполнения одной операции обмена.

Это наиболее простой вид обмена, требующий минимальных затрат технических и программных средств. Однако при его использовании для передачи данных между ЭВМ и сравнительно медленными устройствами, как правило, не удастся полностью загрузить процессор на время между двумя пересылками данных. Так, например, клавишное устройство позволяет организовать ввод данных со скоростью не выше 3-5 символов в секунду, в то время как за 1/5 секунды современный процессор может выполнить несколько сотен миллионов операций.

Асинхронный обмен. При его реализации интервал между командами передачи данных задается самим внешним устройством. Контроллеры этих устройств снабжаются **регистром состояния**, который информирует процессор о готовности устройства к обмену информацией. Программа обмена функционирует следующим образом:

1. проверяется состояние устройства;
2. если устройство готово к обмену, производится переход к п.3. В противном случае повторить п. 1;
3. производится передача очередной порции данных, выполняются действия по их обработке и/или другие операции, и если обмен данными еще не закончен, осуществляется переход к п. 1.

Таким образом, устройство само информирует ЭВМ о завершении предыдущей операции и готовности к выполнению новой операции. Однако процессор должен тратить время на ожидание момента готовности, а так как готовность проверяется программным

путем, в это время процессор не может выполнять никакой другой работы по преобразованию данных.

Обмен по прерыванию программы. Отличается от асинхронного обмена тем, что готовность устройства к обмену проверяется при помощи аппаратных средств ЭВМ, а не программным путем. Это позволяет организовать работу ЭВМ следующим образом:

1. Процессор выполняет какую-либо программу (назовем ее основной или фоновой), не связанную с обменом.
2. Основная программа (через процессор) выдает запрос на выполнение операции обмена (обращение к требуемому порту).
3. Когда устройство готово к приему или выдаче данных, оно посылает в процессор сигнал готовности, воспринимаемый специальным блоком ЭВМ (как правило, **контроллером прерываний**).
4. Контроллер прерываний приостанавливает (прерывает) исполнение основной программы и передает управление подпрограмме, организующей нужный вид обмена данными.
5. Когда выполнение подпрограммы завершается, возобновляется работа процессора по основной (временно прерванной) программе.

Различие между рассмотренными видами программно-управляемого обмена можно проиллюстрировать на следующем примере.

Пример. Необходимо вскипятить воду в чайнике.

Синхронный обмен. В 8 часов чайник с водой поставлен на плиту. Зная, что обычное время закипания такого объема воды 15 мин., ровно в 8ч.15мин. чайник снимается с плиты; вопрос о том, закипела вода или нет не ставится.

Асинхронный обмен. Чайник с водой поставлен на плиту, и через равные промежутки времени, например, раз в 3 минуты, осуществляется проверка, закипела вода или нет. Когда вода закипит, чайник снимается его с плиты.

Обмен по прерыванию. На носик чайника с водой надевается свисток и он (чайник) поставлен на плиту. Пользователь чайника может заниматься своим делом (фоновой работой). Когда свисток засвистит, пользователь снимает чайник с плиты и продолжает фоновую работу.

9. СИСТЕМА ПРЕРЫВАНИЙ

9.1. Основные понятия. Типы прерываний

Прерывание - это сигнал, по которому процессор "узнает" о совершении асинхронного события. Другими словами, прерывание - это ответ вычислительной системы на наступление особого события, нарушающего последовательное выполнение команд текущей программы, то есть, это изменение естественного порядка выполнения программы, которое связано с необходимостью реакции системы на работу внешних устройств, а также на ошибки и особые ситуации, возникшие при выполнении программы.

"Ответ" системы на наступившее событие заключается в запуске программы обработки данного прерывания - **обработчика прерывания**, специальной программы, специфической для каждой возникшей ситуации, после выполнения которой, если это возможно, возобновляется работа прерванной программы.

Замечание. Таким образом, механизм обработки прерываний предоставляет возможность организации ветвления при реализации программных процессов.

В упрощенном представлении можно выделить три типа прерываний:

- 1) внутренние,
- 2) внешние,
- 3) внепроцессорные.

Внутренние прерывания. К прерываниям этого типа относят:

- группу программных прерываний (деление на нуль, переполнение, неверная адресация и т. п.),
- прерывания от схем контроля машины, сбоях системы питания и др.

Обработка прерываний этого типа состоит в выдаче сообщений о причине прерывания, прекращении выполнения текущей программы и перехода к реализации другой программы либо, если дальнейшее функционирование системы невозможно, только в выдаче диагностического сообщения, локализирующего причину отказа.

В любом случае, при наступлении события, вызвавшего аварию, процессор не останавливается.

Внешние прерывания. Эту группу прерываний представляют прерывания от внешних устройств. Обработка событий, связанных с выполнением операций обмена данными между внешними устройствами и ОЗУ, в конечном счете, сводится к запуску *драйвера* - программы, реализующей обмен с устройством конкретного типа (драйвер клавиатуры, драйвер монитора и т. п.).

Внепроцессорные прерывания. Прерывания, обработка которых приводит к передаче управления Общей шиной от процессора к контроллеру внешнего устройства с реализацией дальнейшего обмена между устройством и основной памятью по Общей шине напрямую без посредничества процессора, то есть без запуска какого-либо драйвера (см. п. 10.4.).

9.2 Общая организация прерываний

Механизм прерывания обеспечивается соответствующими аппаратно-программными средствами компьютера.

Задачей *аппаратных средств* обработки прерывания в процессоре ЭВМ является приостановка выполнения одной программы (иногда называемой основной) и передача управления подпрограмме обработки прерывания.

Поскольку для выполнения подпрограммы обработки прерывания используются различные регистры процессора (РОНы, счетчик команд, регистр флагов и т.д.), то информацию, содержащуюся в них в момент прерывания, необходимо сохранить для последующего возврата в прерванную программу.

Обычно задача сохранения содержимого счетчика команд и регистра флагов, содержащего *вектор состояния* процессора возлагается на аппаратные средства обработки прерывания. Сохранение содержимого других регистров процессора, используемых в подпрограмме обработки прерывания, производится непосредственно в подпрограмме (рис. 39.).

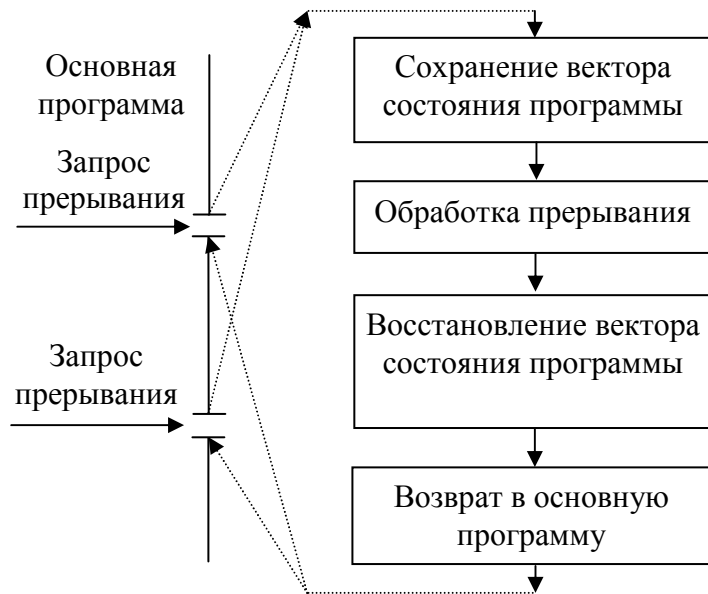


Рис.39. Структура подпрограммы обработки прерывания и ее связь с основной программой

9.3 Организация системы прерываний с использованием векторов прерываний

Рассмотрим подробнее процесс обработки внешних прерываний.

Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к обычной подпрограмме; различие в том, что при обращении к подпрограмме эти действия инициируются командой, а при обработке прерывания - управляющим сигналом от внешнего устройства, называемым **Запрос** (или **Требование**) **прерывания**.

Эта важная особенность обмена с прерыванием программы позволяет организовать обмен данными с внешними устройствами в произвольные моменты времени, не зависящие от программы, выполняемой в ЭВМ. Таким образом, появляется возможность обмена данными с внешними устройствами в реальном масштабе времени, определяемом внешней по отношению к ЭВМ средой (например, с датчиками, следящими за состоянием технологического процесса).

Прерывание программы по требованию внешнего устройства не должно оказывать на прерванную программу никакого влияния, кроме увеличения времени ее выполнения за счет приостановки на время выполнения подпрограммы обработки прерывания.

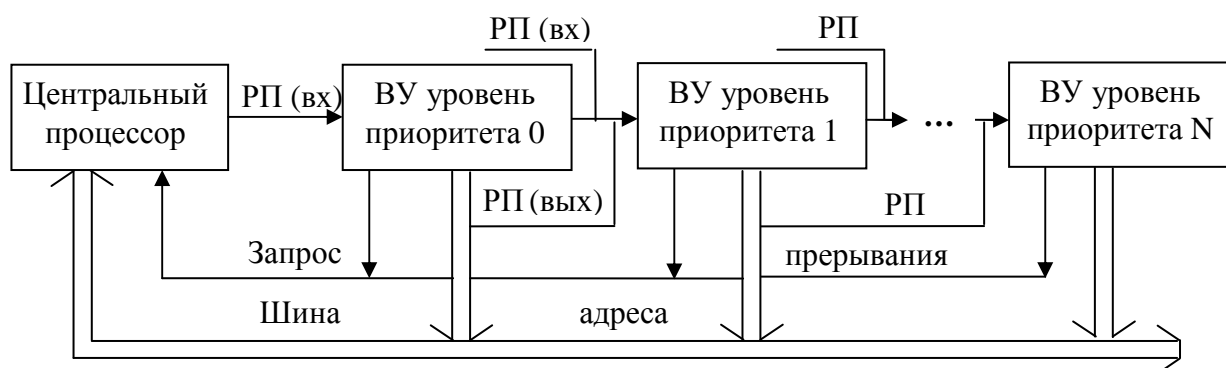
Формирование сигналов прерываний - запросов устройств на обслуживание, происходит в их контроллерах.

В серийных ЭВМ обычно используется одноуровневая система прерываний, то есть сигналы **Запрос прерывания** от всех внешних устройств поступают на один вход процессора. Поэтому возникает проблема идентификации внешнего устройства, запросившего обслуживание. Основным способом, решающим проблему идентификации в большинстве современных ЭВМ в настоящее время, является использование **векторов прерываний**.

Внешнее устройство, запросившее обслуживание, само идентифицирует себя с помощью адреса своего вектора прерывания - ячейки основной памяти, в которой хранится адрес начала программы обработки прерывания данного типа.

Векторы всех обработчиков прерываний собраны в единую **таблицу векторов прерываний**, располагающуюся в самых младших адресах оперативной памяти, имеющую объем 1 Кбайт и содержащую 4-х байтные элементы (векторы прерываний) для 256 обработчиков прерываний. Так как таблица всегда имеет нулевой начальный адрес и длину вектора в 4 байта, чтобы определить адрес вектора для прерывания типа i , достаточно просто умножить это значение на 4.

Вектор прерывания выдается контроллером не одновременно с запросом на прерывание, а только по разрешению процессора (рис.40).



Условные обозначения:

- РП (вх) - Разрешение прерывания (входной);
- РП (вых) - Разрешение прерывания (выходной).

Рис. 40. Реализация приоритетов ВУ в ЭВМ с векторной системой прерывания

В ответ на сигнал контроллера **Запрос прерывания** процессор формирует управляющий сигнал **Разрешение прерывания (входной)**, который разрешает контроллеру, запросившему обслуживание, выдачу адреса своего вектора прерывания в шину адреса системного интерфейса.

Сигнал **Разрешение прерывания (входной)** поступает сначала в контроллер внешнего устройства с наивысшим приоритетом. Если это устройство не требовало обслуживания, его контроллер пропускает сигнал на разрешение прерывания (**Разрешение прерывания (выходной)**) на следующий контроллер, в противном случае дальнейшее распространение сигнала прекращается и контроллер выдает в шину адреса адрес вектор прерывания.

Замечание. На передней панели каждого контроллера расположены два разъема GI и GO для входа и выхода сигнала разрешения прерывания. Согласно протоколу R/G на этой линии действует следующее правило: контроллер, не выставивший запроса на прерывание, передает сигнал разрешения со своего входа GI на свой выход GO, то есть, генерирует сигнал **Разрешение прерывания (выходной)**; контроллер, выставивший запрос, блокирует передачу сигнала разрешения на свой выход GO. Это делается для того, чтобы исключить одновременную выдачу векторов прерывания от нескольких внешних устройств.

Очевидно, что сигнал **Запрос прерывания** выдает контроллер устройства, к которому был выдан программный запрос, то есть, инициатором операции обмена является процессор.

9.4 Организация прямого доступа к памяти

В режиме прямого доступа к памяти (ПДП) обмен данными между внешним устройством и оперативной памятью ЭВМ происходит без участия процессора. Обменом в режиме ПДП управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме ПДП, размещаются в специальном контроллере, который называется **контроллером прямого доступа**.

Режим ПДП обеспечивает возможность использования в ЭВМ быстродействующих внешних запоминающих устройств (как правило, накопителей на магнитных дисках). Это связано с тем, что обмен данными с такими устройствами требует скорости, сравнимой со скоростью обмена с ОЗУ. Другими словами, время на обмен одним байтом данных между

ОЗУ и ВЗУ должно быть равно циклу памяти. Ясно, что достичь такого быстродействия при программно-управляемом обмене не удастся.

При подключении контроллера ПДП возникает проблема совместного использования шин системного интерфейса процессором и контроллером ПДП.

Прямой доступ к памяти с захватом цикла. Наиболее распространенным является способ подключения с "захватом цикла" и принудительным отключением процессора от шин системного интерфейса. Для реализации режима ПДП системный интерфейс ЭВМ дополняется двумя линиями для передачи управляющих сигналов **Запрос прямого доступа к памяти (ЗПДП)** и **Разрешение прямого доступа к памяти (РПДП)**.

Управляющий сигнал **ЗПДП** формируется контроллером ПДП. Процессор, получив этот сигнал, приостанавливает выполнение очередной команды, не дожидаясь ее завершения (!), выдает в системный интерфейс управляющий сигнал **РПДП** и отключается от шины системного интерфейса. С этого момента все шины системного интерфейса управляются контроллером ПДП. Контроллер ПДП, используя шины системного интерфейса, осуществляет обмен одним байтом или словом данных с памятью ЭВМ и затем, сняв сигнал **ЗПДП**, возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим байтом, он вновь "захватывает" цикл процессора и т.д. В промежутках между сигналами **ЗПДП** процессор продолжает выполнять команды программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме прерывания.

Применение в ЭВМ обмена данными в режиме ПДП всегда требует предварительной подготовки, а именно: для каждого внешнего устройства необходимо выделить область памяти, используемую при обмене, и указать ее размер, то есть количество записываемых в память или читаемых из памяти байтов (слов) информации. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байтов (слов). Перед началом обмена в режиме ПДП процессор должен выполнить программу загрузки. Эта программа обеспечивает запись в указанные регистры контроллера ПДП начального адреса выделенной внешнему устройству памяти и ее размера в байтах или словах в зависимости от того, какими порциями ведется обмен, и выполняется в два этапа.

Этап 1. Подготовка внешнего устройства к очередному сеансу обмена. Процессор в режиме программно-управляемого обмена опрашивает состояние внешнего устройства (проверяет его готовность к обмену) и посылает во внешнее устройство команды, обеспечивающие его подготовку к обмену. Такая подготовка может сводиться, например, к перемещению головок на требуемую дорожку в магнитном диске. Затем выполняется загрузка регистров контроллера ПДП. На этом подготовка к обмену в режиме ПДП завершается, и процессор переключается на выполнение другой программы.

Этап 2. Обмен данными в режиме ПДП. Начинается после завершения подготовительных операций по инициативе либо внешнего устройства, как это было рассмотрено выше, либо процессора. В последнем случае контроллер ПДП необходимо дополнить регистром состояния и управления, содержимое которого будет определять режим работы контроллера, анализируемый процессором. Один из разрядов этого регистра будет инициировать обмен данными с внешним устройством. Загрузка информации в регистр состояния и управления контроллера ПДП производится программным путем.

Заметим, что использование в ЭВМ обмена в режиме ПДП требует от программиста очень ясного понимания процессов, происходящих в ЭВМ при выполнении программы, и четкой синхронизации процесса выполнения программы и ввода-вывода в режиме ПДП.

Прямой доступ к памяти с блокировкой процессора. Отличается от ПДП с "Захватом цикла" тем, что управление системным интерфейсом передается контроллеру ПДП не на время обмена одним байтом, а на время обмена блоком данных (например, содержимым сектора магнитного диска). Такой режим ПДП необходим в тех случаях, когда время обмена одним байтом с внешним устройством сопоставимо с циклом процессора. В

этом случае процессор не успевает выполнить хотя бы одну команду между очередными операциями обмена в режиме ПДП.

10. ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ

10.1. Схема взаимодействия устройств в ЭВМ. Операции обмена информацией

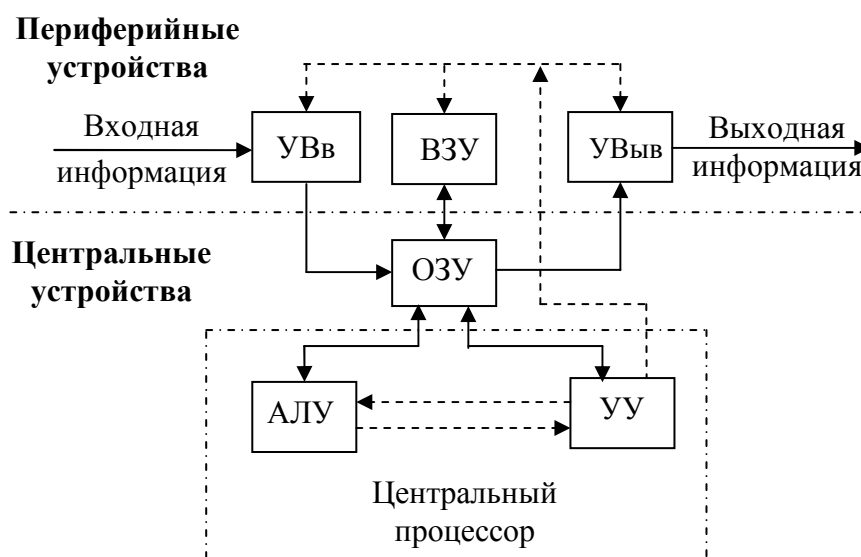
Использование ЭВМ позволяет управлять экспериментами и технологическими процессами, производить сбор данных и выполнять расчеты непосредственно под управлением вычислительной машины, управлять бытовыми приборами и средствами передвижения, автоматизировать делопроизводство и финансовую деятельность, и т.п.

Первоначально словосочетанием "устройства ввода-вывода ЭВМ (УВВ)" обозначалось все внешнее оборудование вычислительных машин, в настоящее время на смену ему пришло определение "периферийное и терминальное оборудование ЭВМ" или "внешние устройства". Это вызвано тем, что все чаще средства ввода-вывода и оперативного взаимодействия с ЭВМ являются не отдельными устройствами, а сложными комплексами.

С помощью периферийных устройств осуществляется связь центральных устройств ЭВМ с различными "поставщиками" и "потребителями" информации. Функции внешних устройств достаточно сложны, однако из них можно выделить две основные:

- хранение информации в том или ином физическом представлении на разных носителях данных,
- преобразование информации соответственно функциям, выполняемым устройством.

На рис. 41. представлена упрощенная схема взаимодействия устройств в ЭВМ. На схеме сплошные линии соответствуют перемещению потоков данных между устройствами, а пунктирные - управляющих сигналов от устройства управления процессора.



Условные обозначения:

УВв – устройства ввода,

УВыв – устройства вывода,

ВЗУ – внешние запоминающие устройства.

Рис. 41. Схема взаимодействия устройств в ЭВМ

Обмен информацией между оперативной памятью и внешними устройствами осуществляется путем использования двух операций:

1. операции ЗАПИСИ (ВЫВОДА) - перенесения информации из оперативной памяти на внешнее устройство,
2. операции ЧТЕНИЯ (ВВОДА) - перенесения информации из внешнего устройства в оперативную память.

Заметим, что по отношению к оперативной памяти операции обмена с внешними устройствами противоположны операциям обмена с центральным процессором (см. п.6.1.)

Так как большинство внешних устройств позволяет человеку общаться с ЭВМ на языке слов и десятичных чисел, а в ЭВМ информация представляется в виде последовательности двоичных чисел (0 и 1), в устройствах ввода-вывода производится кодирование (декодирование) информации, пересылаемой в/из ЭВМ. Существуют различные способы кодирования символьной информации, поступающей из устройств в ЭВМ. Следует помнить, что один внешний символ кодируется комбинацией битов, составляющих один байт информации в памяти ЭВМ.

При обмене данными между внешним устройством и оперативной памятью должен быть учтен важный вопрос о контроле надежности передаваемой в оперативную память информации. Так как внешние устройства имеют достаточно много механических элементов, они менее надежны, чем электронные схемы процессора и памяти. Причиной неправильного восприятия содержимого передаваемого байта могут быть пыль, царапины, масляные пятна и т.п. Короче говоря, при реализации операций обмена между оперативной памятью внешними устройствами могут появиться ошибки. Для обнаружения (а иногда, и исправления) ошибок передаваемые коды преобразуют в какую-либо стандартную форму (например, добавляют еще один бит, чтобы в новом коде было четное число единиц). При передаче такого стандартного кода производится определенного вида контроль (например, контроль четности), и по его результатам принимается решение о правильности принятого (прочитанного) байта информации. Точнее, при записи кода на внешнее устройство содержимое дополнительного бита устанавливается таким образом, чтобы общее количество единиц в сформированном коде было четным (контроль по четности) или нечетным (контроль по нечетности). При чтении байта информации производится соответствующий контроль.

Обоснованием такого подхода к осуществлению контроля является ничтожно малая вероятность искажения более одного бита передаваемого байта информации.

10.2. Назначение и классификация периферийных (внешних) устройств

Периферийное (или *внешнее*) устройство – это устройство, входящее в состав внешнего оборудования микро-ЭВМ, обеспечивающее ввод/вывод данных, организацию промежуточного и длительного хранения данных, передачу информации, но не определяющее архитектуру и принципы функционирования компьютера.

Основное назначение ПУ – преобразование информации, представленной в естественном виде, во внутренний формат компьютера и наоборот.

Классификацию внешних устройств можно вести по разным основаниям:

1. По **назначению** такие устройства можно разделить на следующие группы:
 - устройства для связи "человек - ЭВМ" (устройства ввода, устройства вывода, интерактивные устройства);
 - устройства массовой памяти (внешние запоминающие устройства (ВЗУ) большой емкости);
 - устройства для связи с объектами управления (датчики, реле, устройства преобразования непрерывных сигналов с датчиков в цифровые сигналы и обратного преобразования, и т.д.);
 - средства передачи данных на большие расстояния (средства телекоммуникации).
2. По **выполняемым функциям**:
 - средства ввода-вывода информации,

- средства хранения информации,
- средства телеобработки (коммутации и приема-передачи информации).

На рис. 42. представлены основные функциональные классы периферийных устройств.

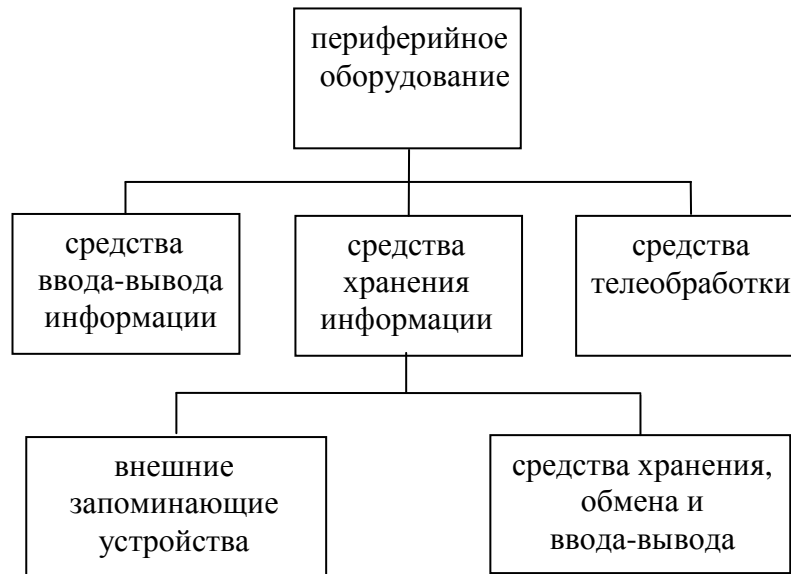


Рис. 42. Классификация периферийного оборудования

В средствах хранения информации выделение второй группы объясняется двойным характером использования некоторых типов устройств в составе внешнего оборудования и персональных компьютеров: и как внешних запоминающих устройств, и как устройств ввода-вывода.

Конкретное рассмотрение различных типов периферийных устройств начнем с внешних запоминающих устройств большой емкости.

10.3. Внешние запоминающие устройства большой емкости

Внешние запоминающие устройства обеспечивают большой объем памяти и относительно недороги, но обладают меньшим быстродействием, чем основная память ЭВМ. Почти все такие запоминающие устройства предназначены для долговременного хранения программ и данных.

10.3.1. Накопители с жесткими магнитными дисками.

10.3.1.1. Организация жестких дисков

Обычно накопители с жесткими магнитными дисками (НЖМД) в России называют «Винчестером» (на Западе это название практически не используется). По одной из версий при его разработке инженеры использовали краткое внутреннее название «30-30», что означало два модуля (в максимальной компоновке) по 30 мегабайт каждый, что по созвучию совпало с обозначением популярного охотничьего оружия — винтовки Winchester Model 1894, использующего винтовочный патрон 30-30.

Замечание. Различают *съёмные* и *несъёмные* НЖМД. Несъёмный диск является системным устройством, а съёмный – периферийным. Организация обоих типов этих устройств абсолютно идентична.

Жесткий диск состоит из одной или нескольких стеклянных или металлических пластин, каждая из которых покрыта с одной или двух сторон магнитным материалом, помещённых в коробку.

Таким образом, диск в общем случае состоит из **пакета** пластин (рис.43.).

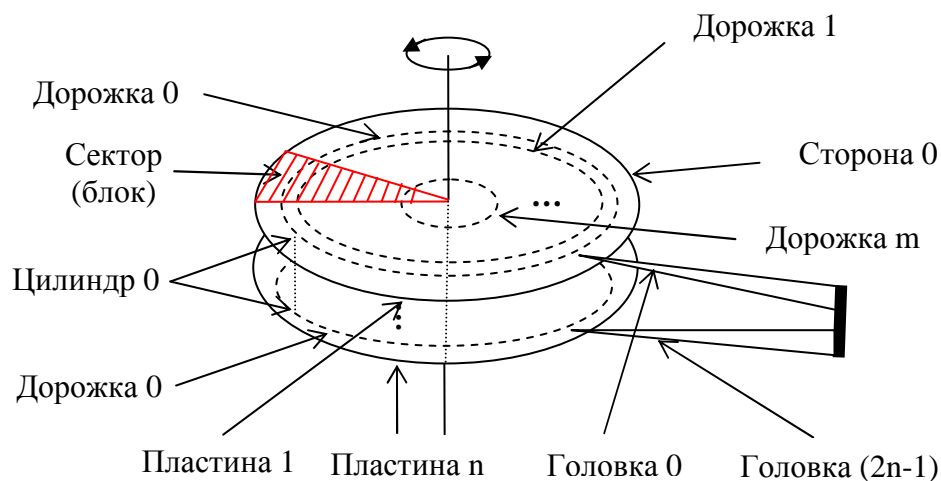


Рис.43. Схема устройства жесткого диска

На каждой стороне каждой пластины размечены тонкие концентрические кольца — **дорожки**, на которых хранятся данные. Количество дорожек зависит от типа диска. Нумерация дорожек начинается с 0 от внешнего края к центру диска. Когда диск вращается, элемент, называемый **головкой**, считывает двоичные данные с магнитной дорожки или записывает их на магнитную дорожку.

Головка может позиционироваться над заданной дорожкой. Головки перемещаются над поверхностью диска дискретными шагами, каждый шаг соответствует сдвигу на одну дорожку. Запись на диск осуществляется благодаря способности головки изменять магнитные свойства дорожки. В некоторых дисках вдоль каждой поверхности перемещается одна головка, а в других — имеется по головке на каждую дорожку. Обычно все головки закреплены на едином перемещающем механизме и двигаются синхронно. Поэтому, когда головка фиксируется на заданной дорожке одной поверхности, все остальные головки останавливаются над дорожками с такими же номерами. В тех же случаях, когда на каждой дорожке имеется отдельная головка, никакого перемещения головок с одной дорожки на другую не требуется, за счет этого экономится время, затрачиваемое на поиск данных.

При вращении пакета головки чтения/записи как бы скользят по поверхности **цилиндра**, образованного дорожками с одинаковыми номерами рабочих поверхностей всех пластин. Другими словами, цилиндр - это совокупность дорожек одного радиуса на всех поверхностях всех пластин пакета.

Каждая дорожка разбивается на фрагменты, называемые **секторами**. Сектор имеет фиксированный для конкретной системы размер, выражающийся степенью двойки, чаще всего размер сектора составляет 512 байт. Учитывая, что дорожки разного радиуса имеют разную длину, возможно размещение на разных дорожках различного количества секторов. Поэтому в современных дисковых устройствах дорожки (цилиндры) делятся на зоны (рис.44.).

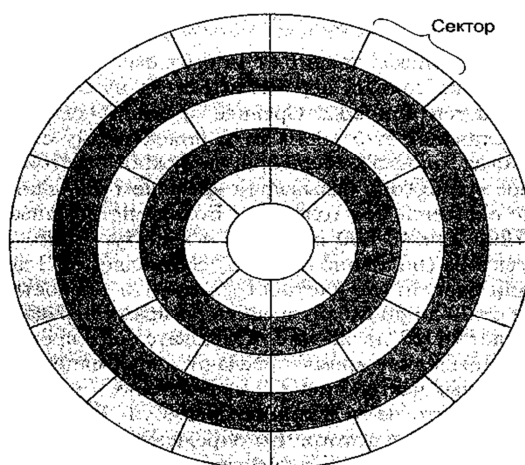


Рис.44. Схема деления цилиндров на зоны

В современных винчестерах количество зон от колеблется от 8 до 20.

На дорожках каждой зоны количество секторов одинаково, однако при продвижении от центра диска число секторов на дорожке в каждой зоне возрастает. Это усложняет процедуру хранения информации на дорожке, но зато повышает емкость диска, что считается более важным.

Сектор — наименьшая **адресуемая единица обмена** данными дискового устройства с оперативной памятью. Для того чтобы контроллер мог найти на диске нужный сектор, необходимо задать ему все составляющие адреса сектора: **номер поверхности, номер цилиндра (дорожки) и номер сектора**.

Не все секторы используются в качестве рабочих. Часть секторов являются запасными. При первоначальной разметке дисков на заводе-изготовителе производится проверка поверхности диска, и информация об обнаруженных дефектных участках записывается в таблицу дефектов, которая размещается в инженерной зоне. В процессе функционирования винчестера эта таблица используется для переназначения (переадресации) обращения к дефектным участкам (секторам) на обращение к хорошим секторам, которые как раз и размещаются на запасных дорожках. Ввиду важности служебной информации инженерная зона различных моделей накопителей может содержать от 2 до 6 копий,

Многие производители указывают размер неформатированного (неразмеченного на сектора) диска, как будто каждая дорожка содержит только данные. В действительности, каждый сектор несет не только данные, но и служебную информацию:

1. В начале каждого сектора записывается его заголовок (prefix), включающий идентификатор (ID) (информацию о номере цилиндра, головки и сектора по которому определяется начало и номер сектора), первую CRC (контрольную сумму) и интервал включения записи, в конце - заключение (suffix), в котором находится контрольная сумма (checksum, CRC), необходимая для проверки целостности данных. Интервал включения записи, после которого следует 512 байт данных.
2. За данными располагается вторая CRC.
3. Интервал между записями (секторами), необходимый для того, чтобы застраховать следующий сектор от записи на предыдущий. Это может произойти из-за неравномерной скорости вращения диска.
4. Преиндексный интервал, служащий для компенсации неравномерности скорости вращения диска.

Емкость форматированного диска обычно на 15% меньше емкости неформатированного.

Очевидно, что при обмене с информацией, расположенной на одном цилиндре, не требуется физическое перемещение головок чтения/записи, что ускоряет выполнение операций обмена. Несмотря на то, что время доступа в магнитных дисках зависит от местоположения головок, оно не зависит от взаимного расположения других порций информации, размещенных на устройстве. Поэтому магнитные диски относятся к классу внешних устройств *с прямым доступом*. Подавляющее большинство остальных периферийных устройств являются устройствами *с последовательным доступом*, в которых для доступа к требуемой порции информации необходим просмотр всех предыдущих порций (от начала или конца носителя).

10.3.1.2. Характеристики жестких дисков

Общие характеристики

- **Интерфейс** — набор, состоящий из линий связи, сигналов, посылаемых по этим линиям, технических средств, поддерживающих эти линии, и правил обмена.
- **Физический размер (форм-фактор)** - современные накопители для персональных компьютеров и серверов имеют размер либо 3.5", либо 2.5". Последние чаще применяются в ноутбуках.
- **Надёжность** — определяется как среднее время наработки на отказ.
- **Сопrotивляемость ударам** — сопротивляемость накопителя резким скачкам давления или ударам, измеряется в единицах допустимой перегрузки во включённом и выключенном состоянии.
- **Скорость вращения шпинделя** — количество оборотов шпинделя в минуту. От этого параметра в значительной степени зависят время доступа и скорость передачи данных. В настоящее время выпускаются винчестеры со следующими стандартными скоростями вращения: 4200, 5400 и 7200 (ноутбуки), 7200 и 10 000 (персональные компьютеры), 10000 и 15000 об./мин. (серверы и высокопроизводительные рабочие станции). Увеличению скорости вращения шпинделя в винчестерах для ноутбуков препятствует гироскопический эффект, влияние которого пренебрежимо мало в неподвижных компьютерах.

Производительность диска зависит от следующих временных характеристик:

1. **Время поиска t_{sk}** – время, требуемое для перемещения с произвольной дорожки до заданной дорожки, но без готовности чтения; среднее время поиска между дорожками, взятыми наугад, составляет от 5 до 10 мс, а поиск между смежными дорожками — менее 1 мс.
2. **Время установки головки t_{set}** – время, необходимое для стабилизации вибраций головки в конце этапа поиска.
3. **Время задержки из-за вращения диска t_r (время ожидания сектора)** – это время, требуемое головке чтения для поворота от произвольного сектора до требуемого сектора на той же дорожке. Большинство жестких дисков, используемых в современных компьютерах, имеют скорость вращения 7200 об/мин, в ноутбуках часто применяются накопители со скоростью 5400 об/мин. Таким образом, типовое время задержки из-за вращения диска составляет около 4 мс.
4. **Время произвольного доступа t_a** – среднее время, за которое винчестер выполняет операцию позиционирования головки чтения/записи на произвольный участок магнитного диска, то есть время, необходимое для перемещения головки чтения с текущей дорожки до начала считывания данных из заданного сектора. Представляет собой сумму трех величин:

$$t_a = t_{sk} + t_{set} + t_r$$

Как правило, минимальным временем обладают диски для серверов, самым большим — диски для портативных устройств. Для сравнения, у **SSD-накопителей** этот параметр меньше 1 мс.

5. **Время передачи информации** зависит от плотности записи и скорости вращения.

Характеристики организации компонентов хранения информации

1. **Форматированная емкость** – это объем накопителя, доступный для хранения пользовательских данных после форматирования накопителя. Ёмкость жёстких дисков (с форм-фактором 3,5 дюйма) близится к 5 Тб. При обозначении емкости жестких дисков производители используют величины, кратные 1000, т.е. 1 Гб = 1,000,000,000 байт.
2. **Линейная плотность записи** определяет количество битовых ячеек на единицу длины дорожки жесткого диска и обычно измеряется в битах/дюйм (рис.45.).

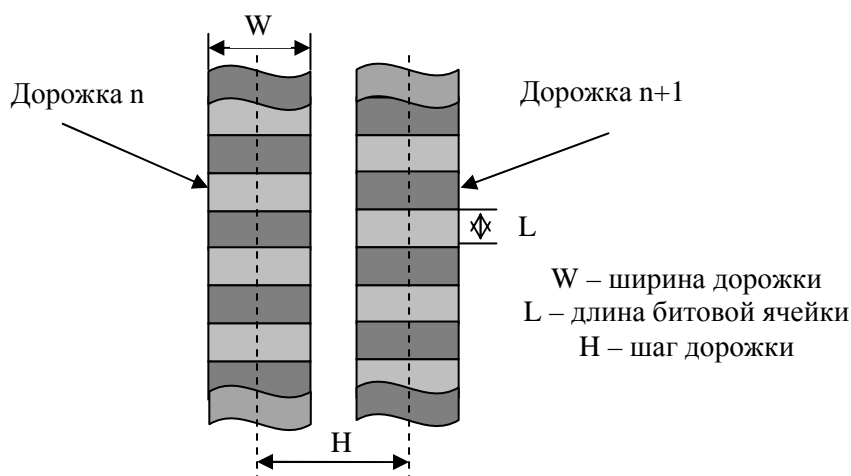


Рис.45. Компоненты хранения информации

Под **шагом дорожек** подразумевают расстояние между центрами смежных дорожек.

Ширина дорожки зависит от величины головки и от точности ее перемещения. Размеры головок чрезвычайно малы, у современных накопителей они имеют порядок 30-50 нм, что в несколько тысяч раз меньше толщины человеческого волоса.

3. **Плотность дорожек** зависит от ширины и шага дорожек и определяет количество дорожек на пластине. Измеряется в дорожках на дюйм.
4. **Поверхностная плотность записи** определяется как величина, обратная площади битовой ячейки, и имеет размерность Гб/дюйм². Очевидно, что поверхностная плотность записи является произведением линейной плотности и плотности дорожек. Одна из основных целей исследований магнитной записи – повышение одновременно линейной плотности и плотности дорожек для достижения максимальной поверхностной плотности записи. Повышение поверхностной плотности записи приводит к увеличению емкости накопителя, сокращению удельной стоимости хранения данных и уменьшению времени доступа.

Очевидно, что приводимые характеристики геометрии диска напрямую зависят от расстояния между головкой и диском. Аналогичный эффект происходит при чтении: чем ближе текст поднести к глазам, тем меньший размер шрифта доступен для распознавания. В современных дисках это расстояние составляет около 10 нм (1нм = 10⁻⁹м).

Важнейшее условие достижения такой величины зазора (кроме наличия соответствующих технологий) – полное отсутствие влияния внешней среды (пыль, влага). Поэтому винчестеры герметично закрываются в вакуумную упаковку.

Некоторые из остальных важных характеристик

1. **Скорость передачи данных** – количество записанных/считанных бит в единицу времени, которые обрабатываются головкой. Она зависит от линейной плотности записи и скорости вращения дисков и различается на внешнем и внутреннем радиусе пластины.
2. **Объём буфера** - промежуточной памяти, предназначенной для сглаживания различий скорости чтения/записи и передачи по интерфейсу. Варьируется от 8 до 128 Мб. При огромных скоростях вращения пластин диска и высокой плотности записи информации контроллер жёсткого диска не успевает обрабатывать читаемую головками информацию (внутренняя скорость чтения велика), поэтому данные первоначально сбрасываются в локальную оперативную память диска, откуда затем постепенно извлекаются контроллером.

Замечание. Данный буфер не следует ассоциировать с [Кэшем диска](#).

При выборе жесткого диска для обычного настольного ПК полезно ориентироваться на такие потребительские качества, как:

- **тишина работы** (низкий акустический шум),
- **малое энергопотребление** (нагрев) в работе,
- **хорошая ударостойкость**,
- **надежность**.

Последний, очень важный критерий, потребитель обычно выясняет по информации от более опытных потребителей, потому что фактически не существует объективных независимых источников с открытой и достаточно достоверной статистикой.

По остальным критериям выбора возможно ориентироваться объективно.

10.3.1.3. Методы записи информации на диск

Метод продольной записи, при использовании которого биты информации записываются с помощью маленькой головки, которая, проходя над поверхностью вращающегося диска, намагничивает миллиарды горизонтальных дискретных областей - доменов. При этом вектор намагниченности домена расположен продольно, т.е. параллельно поверхности диска. Каждая из этих областей является логическим нулём или единицей, в зависимости от намагниченности. Максимально достижимая при использовании данного метода плотность записи составляет около 23 Гбит/см². В настоящее время происходит постепенное вытеснение данного метода более прогрессивными технологиями.

Метод перпендикулярной записи, при котором биты информации сохраняются в вертикальных доменах. Это позволяет использовать более сильные магнитные поля и снизить площадь материала, необходимую для записи 1 бита. Плотность записи у современных образцов - более 60 Гбит/см². Жёсткие диски с перпендикулярной записью доступны на рынке с 2005 года.

Главное отличие между данными технологиями заключается в направлении намагниченности доменов - в случае параллельной записи оно параллельно плоскости диска, а в случае перпендикулярной, соответственно, перпендикулярно.

Причина более высокой плотности перпендикулярной записи объясняется не какими-то внутренними характеристиками одного домена, а силами взаимодействия между соседними ячейками.

Среди *экспериментальных типов* записи можно отметить:

Метод тепловой магнитной записи или *термоассистируемую магнитную запись*, самую перспективную и активно разрабатываемую. При использовании этого метода используется точечный подогрев диска, который позволяет головке намагничивать очень мелкие области его поверхности. После того, как диск охлаждается, намагниченность "закрепляется". В экспериментальных образцах, плотность записи достигает 150 Гбит/см². Эксперты оценивают предел плотности записи от 2,3 ÷ 3,1 Тбит/см² до 7,75 Тбит/см².

Структурированные носители. В современных накопителях каждый магнитный домен состоит из нескольких десятков (70-100) мелких структурных элементов ("зерен"), каждое из которых теоретически способно выполнять функции домена и содержать в себе 1 бит информации. В результате появляется возможность увеличить размеры отдельного "зерна" и хранить единицу информации в меньшем количестве "зерен". Если же разработчикам удастся придумать материалы с однозернистыми "островами", то возможны результаты с плотностью до нескольких десятков и даже сотен терабит.

Несмотря на то, что структурированная и термоассистируемая записи абсолютно различны, теоретически эти методы не противоречат друг другу. Другими словами, в будущем возможно появление накопителей, сочетающих оба подхода.

Одной из главных сложностей разработки и структурированных и термоассистируемых носителей является дисковый материал, то есть разработка доступного вещества, удовлетворяющего требованиям обеих технологий.

Концептуальная идея таких носителей крайне проста, однако перспективы ее практической реализации не понятны.

10.3.2. Оптические диски

Оптические диски относятся к типу немагнитных внешних запоминающих устройств. Запоминание и поиск информации в этих устройствах реализуется оптическими средствами. Используемая техника основана на применении полупроводниковых лазеров и оптических систем, которые генерируют очень маленькую световую точку, фокусируемую на тонком слое среды для выборки бита информации; интенсивность отраженного луча соответствует значениям 0 или 1.

Оптические запоминающие устройства обладают более высокой плотностью записи, чем магнитные устройства (за счет слежения за дорожкой и ее подстройкой оптической системы), и не требуют плотного контакта между носителем и считывающей головкой как в магнитном устройстве. Информация на оптическом диске записывается на одну спиралевидную дорожку (как на грампластинке), содержащую чередующиеся участки с различной отражающей способностью.

Носитель в оптическом устройстве имеет неровную поверхность. Чувствительный слой находится под прозрачным защитным покрытием. Световой луч не фокусируется на наружной поверхности, поэтому пыль и царапины на поверхности не имеют никакого значения.

Наиболее распространенными являются два типа носителей информации для оптических запоминающих устройств:

1. **Дисковод CD** (Compact Disk - компакт-диск),
2. **Дисковод DVD** (Digital Video Disk, цифровой видеодиск или в другой трактовке - Digital Versatile Disk, цифровой многоцелевой диск) является дальнейшим развитием лазерных технологий.

Изначально компакт-диск был создан для хранения аудиозаписей в цифровом виде (известен как CD-Audio), однако в дальнейшем стал широко использоваться как носитель для хранения любых данных (файлов) в двоичном виде. Формат файлов на CD-ROM отличается от формата записи аудио-компакт-дисков и потому обычный проигрыватель аудио-компакт-дисков не может воспроизвести хранимую на них информацию, для этого требуется специальный привод (устройство) для чтения таких дисков (сейчас имеются практически в каждом компьютере).

Компакт-диск (CD-ROM) стал основным носителем для переноса информации между компьютерами (вытеснив с этой роли флоппи-диски – накопители на гибких магнитных дисках). Сейчас он уступает эту роль более перспективным твердотельным носителям.

Формат DVD по структуре данных бывают четырёх типов:

- DVD-видео — содержат фильмы (видео и звук);
- DVD-Audio — содержат аудиоданные высокого качества (гораздо выше, чем на аудио-компакт-дисках);
- DVD-Data — содержат любые данные;
- смешанное содержимое.

В отличие от компакт-дисков, в которых структура аудиодиска принципиально отличается от диска с данными, в DVD используется одна файловая система UDF (возможно, с рядом ограничений — например, не допускается фрагментация файлов). Таким образом, любой из типов носителей DVD может нести любую из четырёх структур данных.

Физически DVD может иметь одну или две рабочие стороны и один или два рабочих слоя на каждой стороне. От их количества зависит ёмкость диска (из-за чего 8-см диски получили названия DVD-1, -2, -3, -4, а 12-см диски — DVD-5, -9, -10, -14, -18, по принципу округления ёмкости диска в Гб до ближайшего сверху целого числа)

Таким образом, DVD-диски имеют гораздо большую информационную емкость по сравнению с CD-дисками за счет:

1. использования лазеров с меньшей длиной волны, что позволяет размещать оптические дорожки более плотно,
2. информация на DVD-дисках может быть записана на двух сторонах, причем в два слоя на одной стороне.

При работе CD и DVD устройств используется **красный** лазерный луч. Оптические устройства, основанные на работе с **синим** лазером, имеющим меньшую длину волны, чем красный, имеют значительно большую емкость (HP DVD и Blu-Ray).

В настоящее время характеристики стираемых оптических носителей постоянно улучшаются, и эти устройства начинают конкурировать с винчестерскими дисками.

10.3.3. Flash-память

Flash-память - это энергонезависимый тип памяти, позволяющий записывать и хранить данные в микросхемах, помещенных в миниатюрный корпус.

Карты flash-памяти не имеют в своем составе движущихся частей, что обеспечивает высокую сохранность данных при их использовании в мобильных устройствах.

Твердотельные накопители SSD

SSD (*solid-state drive*) — компьютерное немеханическое запоминающее устройство на основе микросхем памяти.

Различают два вида твердотельных накопителей: основанные на флеш-памяти (NAND SSD) и основанные на оперативной памяти (RAM SSD).

NAND SSD. Накопители, построенные на использовании *энергонезависимой* памяти.

До недавнего времени существенно уступали традиционным накопителям — жестким дискам — в скорости записи, но компенсировали это высокой скоростью поиска информации (начального позиционирования). Сейчас уже выпускаются твердотельные накопители со скоростью чтения и записи, во много раз превосходящие возможности жестких дисков. Характеризуются относительно небольшими размерами и низким энергопотреблением.

RAM SSD. Накопители, построенные на использовании *энергозависимой* памяти (такой же, какая используется в **ОЗУ** персонального компьютера); характеризуются сверхбыстрым чтением, записью и поиском информации. Основным их недостатком

является чрезвычайно высокая стоимость. Используются, в основном, для ускорения работы крупных систем управления базами данных и мощных графических станций. Такие накопители, как правило, оснащены аккумуляторами для сохранения данных при потере питания, а более дорогие модели — системами резервного и/или оперативного копирования.

Преимущества

1. Отсутствие движущихся частей, отсюда:

- Полное отсутствие шума (уровень шума — 0 дБ);
- Высокая механическая стойкость;
- Стабильность времени считывания файлов вне зависимости от их расположения или фрагментации;
- Высокая скорость чтения/записи, нередко превосходящая пропускную способность интерфейса жесткого диска;
- Скорость выполнения операций (IOPS – количество операций ввода-вывода в секунду) выше в десятки тысяч раз, чем у жесткого диска.
- Низкое энергопотребление;
- Широкий диапазон рабочих температур;
- Большой модернизационный потенциал, как у самих накопителей, так и у технологий их производства;

2. Отсутствие магнитных пластин, отсюда:

- Намного меньшая чувствительность к внешним электромагнитным полям;
- Малые габариты и вес (нет необходимости делать увесистый корпус для экранирования).

Недостатки

- Главный недостаток NAND SSD — ограниченное количество циклов перезаписи. Обычная флеш-память позволяет записывать данные примерно 10 000 раз. Более дорогостоящие виды памяти — около 100 000 раз. Для борьбы с неравномерным износом применяются схемы балансирования нагрузки. Контроллер хранит информацию о том, сколько раз какие блоки перезаписывались и при необходимости «меняет их местами». — Данный недостаток отсутствует у RAM SSD.
- Проблема совместимости SSD накопителей с устаревшими и даже многими актуальными версиями ОС семейства Microsoft Windows, которые не учитывают специфику SSD накопителей и дополнительно изнашивают их (например, использование операционными системами механизма свопинга (подкачки) на SSD с большой вероятностью, уменьшает срок эксплуатации накопителя;
- Цена гигабайта SSD-накопителей существенно выше цены гигабайта накопителя на жёстких магнитных дисках - HDD. К тому же, стоимость SSD прямо пропорциональна их ёмкости, в то время как стоимость традиционных жёстких дисков зависит не только от количества пластин и медленнее растёт при увеличении объёма накопителя.
- Применение в SSD-накопителях команды TRIM делает невозможным восстановление удалённой информации системными утилитами.
TRIM — команда, позволяющая операционной системе уведомить твердотельный накопитель о том, какие блоки данных уже не содержатся в файловой системе и могут быть использованы накопителем для физического удаления.
- Невозможность восстановить информацию при перепаде напряжения. Так как контроллер и носитель информации в SSD находятся на одной плате, то при превышении или перепаде напряжения чаще всего сгорает весь SSD носитель с безвозвратной гибелью информации. Напротив, в жёстких дисках чаще сгорает

только плата контроллера, что делает возможным восстановление информации с приемлемой трудоёмкостью. Вообще, если произошёл аппаратный отказ SSD из-за выхода из строя чипа контроллера или флеш-памяти, это делает процесс восстановления информации практически неосуществимым.

В настоящее время твердотельные накопители используются в компактных устройствах: ноутбуках, коммуникаторах и смартфонах, но могут быть использованы и в стационарных компьютерах для повышения производительности.

Существуют и так называемые гибридные жесткие диски, появившиеся, в том числе, из-за текущей, пропорционально более высокой стоимости твердотельных накопителей. Такие устройства сочетают в одном устройстве HDD и твердотельный накопитель относительно небольшого объёма, в качестве КЭШа (для увеличения производительности и срока службы устройства, снижения энергопотребления).

10.3.4. Стримеры

Стримеры — это накопители на магнитной ленте. Их отличает сравнительно низкая цена. К недостаткам стримеров относят малую производительность (она связана, прежде всего, с тем, что магнитная лента — это устройство последовательного доступа) и недостаточную надежность (кроме электромагнитных наводок, ленты стримеров испытывают повышенные механические нагрузки и могут физически выходить из строя).

10.4. Средства ввода-вывода информации

Среди устройств ввода-вывода информации можно выделить:

- средства обмена с документами,
- средства непосредственного взаимодействия с ЭВМ.

Параметрами классификации устройств обмена с документами являются:

- тип информации (текстовый, графический),
- функциональное назначение (ввод, вывод),
- степень автоматизации процесса ввода,
- тип носителя информации (печатный документ, электронный документ).

Основным параметром классификации средств непосредственного взаимодействия с ЭВМ является способ взаимодействия (рис.46).

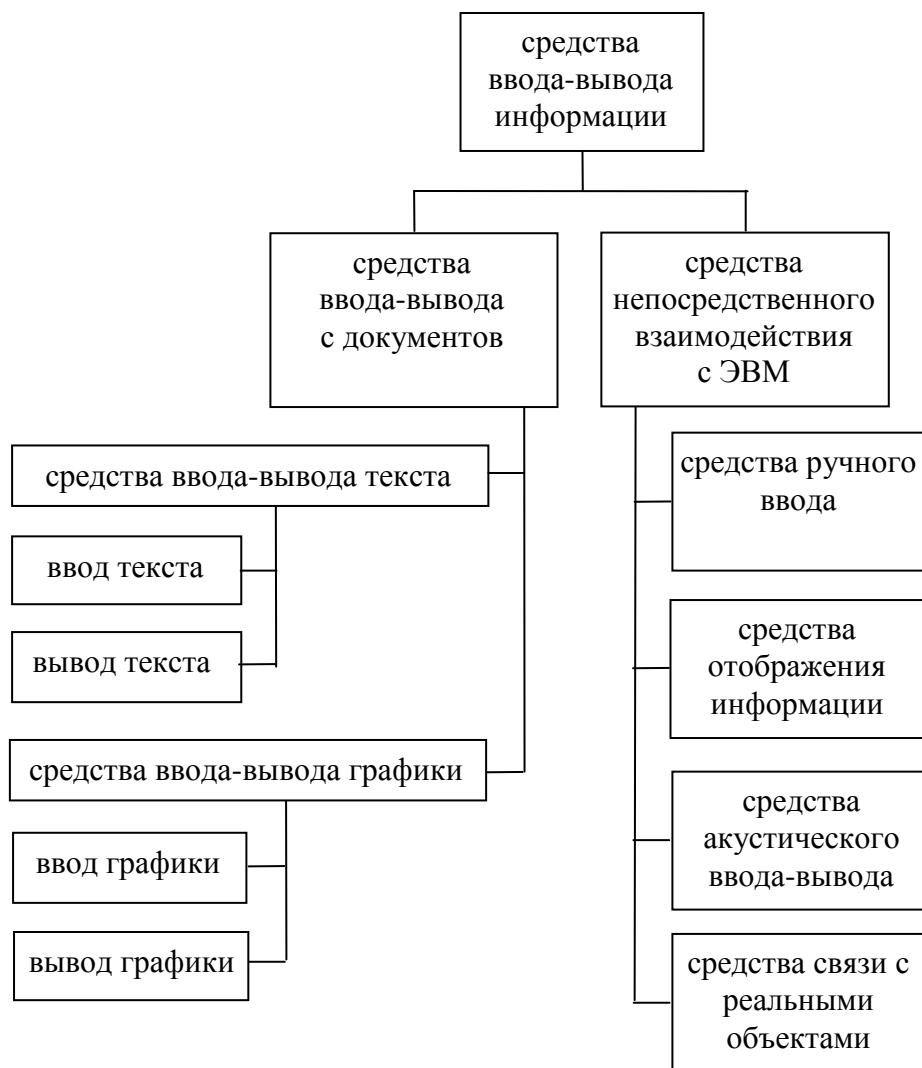


Рис. 46. Классификация средств ввода/вывода

Основная совокупность устройств связи с пользователями представлена в следующей таблице.

Устройства вывода	Устройства ввода	Интерактивные устройства
На бумажные носители: <ul style="list-style-type: none"> • принтер, • плоттер. На электронные носители: <ul style="list-style-type: none"> • монитор Синтезаторы звука	Устройства ввода символьных данных: <ul style="list-style-type: none"> • клавиатура, • специальные клавиатуры. Устройства ввода графических данных: <ul style="list-style-type: none"> • сканер, • графические планшеты (диджитайзеры), • цифровые видеокамеры. Устройства командного управления (манипуляторы): <ul style="list-style-type: none"> • мышь, • трекбол, • джойстик, • тачпад, 	Терминалы ЖК-планшет с сенсорным вводом Интерактивная доска Интерактивный проектор Интерактивная приставка

	<ul style="list-style-type: none"> • пенмаус, • световое перо. Устройства ввода звуковой информации	
--	---	--

10.4.1. Устройства ввода

10.4.1.1. Устройства ввода символьной информации

Клавиатура - важнейшее из устройств ввода. Подавляющее большинство современных клавиатур являются полноходовыми контактными, т.е. клавиша утапливается при нажатии и замыкает контакт между двумя металлическими пластинками, покрытыми, во избежание окисления, пленкой благородного металла. Хорошая клавиатура рассчитана на несколько десятков миллионов нажатий каждой клавиши.

Как правило, внутри корпуса любой клавиатуры, кроме датчиков клавиш, расположены электронные схемы дешифрации и микроконтроллер.

Основной принцип работы клавиатуры заключается в сканировании переключателей клавиш. Замыканию и размыканию любого из этих переключателей соответствует уникальный цифровой код - скан-код. При нажатии клавиши генерируется связанный с ней код, заносимый в соответствующий буфер памяти, а при ее отпуске – другой код, что позволяет перепрограммировать назначение клавиш, вводя новую таблицу соответствия этих кодов.

Ряд клавиш при совместном нажатии пары из них генерируют специальный код, отличный от того, который генерируется при нажатии каждой клавиши в отдельности. Это позволяет значительно увеличить возможности клавиатуры. Вспомним, что для передачи всех возможностей при байтовой системе кодирования могло бы понадобиться 256 клавиш, чего нет ни на одной клавиатуре благодаря указанным совмещениям.

Большинство клавиатур имеют стандартные группы клавиш:

- клавиши пишущей машинки – для ввода букв, цифр и других знаков;
- служебные клавиши, перенацеливающие действия остальных (переключатели регистров, переходы с латинского шрифта на русский и другие);
- функциональные клавиши F1 – F12 (иногда их меньше), назначение которых задает разработчик прикладной программы;
- дополнительные цифровые клавиши для большего удобства в работе.

Важным свойством клавиатуры, благодаря которому пользователь может работать не один час подряд, является *эргономичность*. Этим термином задается совокупность характеристик, определяющих удобство (в широком смысле слова) устройства. По отношению к клавиатуре это:

- дизайн,
- отсутствие бликов,
- удобное взаиморасположение и размеры клавиш,
- и многое другое.

Специальные клавиатуры. Клавиатуры, имеющие специальную форму, рассчитанную с учетом требований эргономики, называют *эргономичными* клавиатурами. Их целесообразно применять на рабочих местах, предназначенных для ввода большого количества знаковой информации. На практике подобными клавиатурами оснащают только специализированные рабочие места.

По методу подключения к системному блоку различают **проводные** и **беспроводные** клавиатуры. Передача информации в беспроводных системах осуществляется инфракрасным лучом. Обычный радиус действия таких клавиатур составляет несколько метров.

10.4.1.2. Устройства ввода графических данных

Сканеры. Принцип действия и классификация сканеров

Сканирование — это процесс оцифровки изображения, иными словами, перевод его в компьютерный вид.

Сканер позволяет вводить в компьютер образы изображений, представленных в виде текста, рисунков, слайдов, фотографий или другой графической информации.

Принцип действия

Сканируемый объект кладется на стекло планшета сканируемой поверхностью вниз. Под стеклом располагается подвижная лампа, движение которой регулируется шаговым двигателем. Свет, отраженный от объекта, через систему зеркал попадает на чувствительную матрицу (специализированную аналоговую интегральную микросхему), далее - на алфавитно-цифровой преобразователь (АЦП) и через интерфейс передается в компьютер. За каждый шаг двигателя сканируется полоска объекта; отсканированные образы полосок объединяются программным обеспечением в общее изображение.

Далее изображение сканируется в формат RAW (сырой, необработанный, содержащий полную информацию о хранимом сигнале, не имеющую четкой спецификации), после чего конвертируется в обычный графический формат с применением текущих настроек яркости, контрастности, и т. д. Эта конвертация осуществляется либо в самом сканере, либо в компьютере — в зависимости от модели конкретного сканера.

На параметры и качество RAW-данных влияют такие аппаратные настройки сканера, как время экспозиции матрицы, уровни калибровки белого и черного, и т. п.

Таким образом, сканеры – это, по существу, настольные аналого-цифровые преобразователи. Они превращают аналоговые графические объекты – документы, страницы журнала, фотографии, служебные удостоверения – в цифровые растры, которые распознает система ПК. Соответствующее программное обеспечение позволяет изменять параметры изображения (яркость, контрастность, цветовой тон и использовать спецэффекты).

В настоящее время все сканеры широкого применения – цветные. Имеются сканеры с 32, 36 и даже 42-битным представлением цвета.

Классификация сканирующих устройств может быть проведена по нескольким признакам.

Область применения:

- Сканеры для дома и небольшого офиса. Основная задача таких сканеров - быстрый ввод текста и сканирование несложной графики с приемлемым качеством.
- Бизнес-класс. Эти сканеры, помимо распознавания текста, должны как минимум уметь качественно и быстро оцифровать фотографии образцов продукции или, к примеру, фотоснимки с банкета, посвященного выставке либо дню *фирмы*. Почти все модели допускают установку слайд-приставки и устройства автоматической подачи документов.
- Область работы художников и дизайнеров. К оцифрованному изображению предъявляются достаточно жесткие требования по качеству и достоверности цветопередачи. Интеллект программных средств должен быть также достаточно развит, чтобы обеспечить необходимые корректировки уже в процессе сканирования.
- Сканеры этой категории - рабочий инструмент издателя. Такой сканер должен отличаться безупречно четким изображением, насыщенными, правдивыми цветами и хорошей проработкой в критических областях.
- Сканеры, ориентированные на инженерные задачи, когда работа ведется, как правило, с крупноформатными изображениями. От сканера не требуется

исключительно точная цветопередача, на первом плане - четкость линий, умение отсеять помехи и корректная работа с инженерными пакетами.

- Сканеры, основная задача которых - обеспечить быстрый и качественный ввод больших объемов текстовой информации для представления в электронном виде. Сканеры этого класса ориентированы на банковские, налоговые, государственные структуры, почтовые и транспортные ведомства, а также на другие предприятия с большим документооборотом. Их функциональное оснащение позволяет с высокой скоростью вести обработку больших объемов документов без участия оператора.

Степень прозрачности вводимого оригинала изображения:

- непрозрачные оригиналы (фотографии, рисунки, страницы книг и журналов),
- прозрачные оригиналы (слайды, негативы, пленки).

Тип вводимого изображения:

- черно-белые (штриховые или полутоновые);
- цветные.

Кинематический механизм (способ считывания):

- **Ручные сканеры.** Перемещение оптической системы осуществляется пользователем вручную.

Поэтому неудивительно, что сканирование получается не очень качественное, если дрожит рука. При работе со сканером необходимо медленно (чтобы успевал очиститься буфер текущей строки) и с постоянной скоростью вести его головку вдоль сканируемого документа. Переполнение буфера контролируется светодиодом: в нормальной ситуации он постоянно горит, при заполнении буфера начинает мигать, а при потере информации гаснет, и сканирование следует повторить. Ширина головки ручного сканера невелика (около 100 мм), поэтому для сканирования стандартного листа А4 требуется 2 или 3 прохода. Ручное сканирование – это тяжелая, утомительная физическая работа, которая требует определенных навыков.

Преимуществами ручного сканера является низкая цена, а также возможность сканирования толстых книг, которые другими моделями отсканировать нельзя. Ручные сканеры могут использоваться в комплекте с портативными компьютерами, если нужно сканировать документы в командировке, в полевых условиях, в дороге, в тесноте.

- **Планшетные сканеры** - наиболее популярные и универсальные приборы. Они выглядят как прямоугольная коробочка, под крышку которой помещается лист бумаги формата А4 (иногда большего).

Предназначены для ввода графической информации с прозрачного или непрозрачного листового материала и обеспечивают наибольшую точность сканирования, так как документ при сканировании неподвижен, а оптическая система осуществляет прецизионное движение. На таких сканерах можно сканировать страницы журналов и не очень толстых книг. В случае толстых книг качество сканирования падает (как и в ксероксах), так как не удается обеспечить плотное прилегание сканируемого листа вблизи корешка книги.

К недостаткам планшетных сканеров следует отнести то, что приходится укладывать сканируемые листы вручную.

Основными потребительскими параметрами планшетных сканеров являются:

- ✓ разрешающая способность;
- ✓ производительность;
- ✓ динамический диапазон;
- ✓ максимальный размер сканируемого материала.

- **Листовой сканер** *позволяет* обрабатывать только один лист, протягивая его между барабанами. Принцип работы схож с работой сканера, так как изображение равномерно протягивается через сканер посредством специального механизма. Данный вид – нечто среднее между ручным и планшетным сканером. По сравнению с первым видом сканеров, протяжной вид позволяет получать более высокое качество изображения, также удобен в использовании, по сравнению с планшетными моделями – более компактные размеры. Такой тип сканеров иногда интегрируется в современное МФУ (многофункциональное устройство – сканер-копир-принтер-факс).
- **проекционные сканеры** - вводимый документ кладется на поверхность сканирования изображением вверх, при этом блок сканирования также находится сверху, а перемещается только сканирующее устройство (возможно сканирование проекций трехмерных предметов).
- **штрих-сканеры**. Эта разновидность ручных сканеров предназначена для ввода данных, закодированных в виде штрих-кода. Такие устройства имеют применение в розничной торговой сети.

Тип используемого интерфейса

Графический планшет, диджитайзер (graphics tablet, digitizer - оцифрователь) — это устройство для ввода графических данных от руки непосредственно в компьютер. Диджитайзер (или дигитайзер) состоит из пера и плоского планшета, чувствительного к нажатию или близости пера. Также может прилагаться специальная мышь.

Устройство может быть основано на разных технических принципах, но при использовании любого из них контур изображения обводится специальным пером. Диджитайзер достаточно редко используемое периферийное устройство, однако, незаменимое в случаях, когда необходимо ввести в компьютер точные данные, представленные в двухмерном виде.

Замечание Часто в качестве устройства ввода упоминают электронное перо, напоминающее обычную школьную ручку. Однако, электронное перо не может работать без графического планшета, так же как сам графический планшет оказывается бесполезной вещью без электронного пера. Поэтому, несмотря на то, что два этих термина существуют отдельно, на самом деле за ними стоит один прибор, состоящий из электронного пера и графического планшета. Их можно рассматривать, как две детали одной системы.

Пенмаус - представляет собой аналог шариковой авторучки, на конце которой вместо пишущего узла установлен узел, регистрирующий величину перемещения.

Световое перо — один из инструментов ввода графических данных в компьютер, разновидность манипуляторов. Внешне имеет вид шариковой ручки или карандаша. Обычно на световом пере имеется одна или несколько кнопок, которые могут нажиматься рукой, удерживающей перо. Ввод данных с помощью светового пера заключается в прикосновениях или проведении линий пером по поверхности экрана монитора. В наконечнике пера устанавливается фотоэлемент, который регистрирует изменение яркости экрана в точке, с которой соприкасается перо, за счёт чего соответствующее программное обеспечение вычисляет позицию, «указываемую» пером на экране и может, в зависимости от необходимости, интерпретировать её тем или иным образом, обычно как указание на отображаемый на экране объект или как команду рисования. Кнопки используются аналогично кнопкам мыши — для выполнения дополнительных операций и включения дополнительных режимов.

Световое перо невозможно использовать с обычными ЖК мониторами. Для этого нужен специальный экран, который может реагировать на световое перо.

Цифровые видеокамеры и фотоаппараты - это устройства, позволяющие получать видеоизображение и фотоснимки непосредственно в цифровом формате. Имеют память, аналогичную компьютерной. Благодаря этому изображения сразу можно вводить в компьютер для хранения на жестком диске, или для преобразования, или для передачи по компьютерным сетям (веб-камера).

Веб-камера (Web-камера) — цифровая видео или фотокамера, способная в реальном времени фиксировать изображения, предназначенные для дальнейшей передачи по сети Интернет.

10.4.1.3. Устройства командного управления

Мышь — механический манипулятор, преобразующий движение в управляющий сигнал. В частности, сигнал может быть использован для позиционирования курсора или прокрутки страниц.

Получила широкое распространение в связи с появлением графического интерфейса пользователя на персональных компьютерах.

Принцип действия

Мышь воспринимает своё перемещение в рабочей плоскости (обычно — на участке поверхности стола) и передаёт эту информацию компьютеру. Программа, работающая на компьютере, в ответ на перемещение мыши производит на экране действие, отвечающее направлению и расстоянию этого перемещения. В разных интерфейсах (например, в оконных) с помощью мыши пользователь управляет специальным курсором — указателем, который является манипулятором элементами интерфейса.

В дополнение к датчику перемещения, мышь имеет одну и более кнопок, а также дополнительные детали управления

Шаровой привод

В шаровом приводе движение мыши передается на выступающий из корпуса обрезиненный стальной шарик (его вес и резиновое покрытие обеспечивают хорошее сцепление с рабочей поверхностью). Два прижатых к шарикам ролики снимают его движения по каждому из измерений и передают их на датчики угла поворота, преобразующие эти движения в электрические сигналы.

Основной недостаток шарового привода — загрязнение шарика и снимающих роликов, приводящее к заеданию мыши и необходимости в периодической её чистке (отчасти эта проблема сглаживалась путём металлизации роликов). Несмотря на недостатки, шаровой привод долгое время доминировал, успешно конкурируя с альтернативными схемами датчиков. В настоящее время шаровые мыши почти полностью вытеснены оптическими мышами.

Оптические мыши с матричным сенсором

В нижней части мыши установлена специальная быстрая видеокамера. Она непрерывно делает снимки поверхности стола и, сравнивая их, определяет направление и величину смещения мышки. Специальная контрастная подсветка поверхности светодиодом или лазером облегчает работу камеры. Оптические мыши работают практически на любых поверхностях, кроме зеркальных или прозрачных.

Пыль и ворс на оптике сенсора могут привести к ошибкам движения или эффекту мелких движений в состоянии покоя, что проявляется дрожанием указателя на экране, иногда с тенденцией сползания в ту или иную сторону.

Некоторые модели оборудуются двумя датчиками перемещения сразу, что позволяет, анализируя изменения сразу на двух участках поверхности, исключать возможные ошибки. Такие мыши иногда способны работать на стеклянных, оргстеклянных и зеркальных поверхностях (на которых не работают другие мыши).

Оптические лазерные мыши

В последние годы была разработана новая, более совершенная разновидность оптического датчика, использующего для подсветки полупроводниковый лазер.

Элементы управления мыши

Кнопки

Кнопки — основные элементы управления мыши, служащие для выполнения основных манипуляций:

- выбора объекта (нажатиями),
- активного перемещения (то есть перемещения с нажатой кнопкой),
- для рисования
- и т. п.).

Количество кнопок на мыши определяет концепция их использования.

Долгое время противостояли друг другу двух- и трёхкнопочные концепции.

Это противостояние закончилось после появления прокрутки экрана (скролла). На двухкнопочной мыши появилась небольшая средняя (третья) кнопка для включения и выключения скроллинга, которая вскоре трансформировалась в колесо прокрутки, нажатие на которое работает как средняя кнопка.

Дополнительные кнопки

Производители постоянно стараются добавить на модели дополнительные кнопки, чаще всего — кнопки под большой или указательный и реже — под средний палец. Некоторые кнопки служат для внутренней настройки мыши (например, для изменения чувствительности) или двойные-тройные щелчки (для программ и игр), другие — для выполнения системных функций, например:

- горизонтальная прокрутка;
- двойное нажатие;
- навигация в браузерах и файловых менеджерах;
- управление уровнем громкости и воспроизведением аудио- и видеоклипов;
- запуск приложений;
- и т. п.

Другие элементы управления

Большинство элементов, не являющихся кнопками, служат для прокрутки (скроллинга) web-страниц, документов, списков, л и т. п.), в окнах приложений и элементах интерфейса. Среди них можно выделить несколько типов.

Колеса и потенциометры — диски, выступающие из корпуса, доступные для вращения. Потенциометры, в отличие от колёс, имеют крайние положения.

Наличие одного колеса между кнопками (или «скролла»; для вертикальной прокрутки) на сегодняшний день является стандартом де-факто.

Такое колесо может отсутствовать у моделей, имеющих для прокрутки иные конструктивы.

Колёса и потенциометры могут быть использованы для регулировки, например, громкости.

Мини джойстик — рычаг с двумя кнопками или сдвоенное под прямым углом плечо, ориентированное в четырёх основных направлениях). Плечо может иметь центральный рычажок или, наоборот, центральное углубление (аналогично джойстикам игровых пультов).

Кроме вертикальной и горизонтальной прокрутки, джойстики мыши могут быть использованы для альтернативного перемещения указателя или регулировок, аналогично колёсам.

Существуют следующие *интерфейсы подключения* мыши.

Беспроводные мыши. Сигнальный провод мыши иногда рассматривается как мешающий и ограничивающий фактор. Этому фактору лишены **беспроводные** мыши. Однако беспроводные мыши имеют серьёзную проблему — вместе с сигнальным кабелем

они теряют стационарное питание и вынуждены иметь автономное, от аккумуляторов или батарей, которые требуют подзарядки или замены, а также увеличивают вес устройства.

Аккумуляторы беспроводной мыши могут подзарядиться как вне мыши, так и внутри неё (точно так же, как аккумуляторы в мобильных телефонах). В последнем случае, мышь должна периодически подсоединяться к стационарному питанию.

Радиосвязь. Наиболее массовым в настоящее время является второе поколение радиомышей. Оно использует свободный частотный диапазон 2,45 ГГц и строится на базе высокоинтегрированных скоростных радиоканалов. Основным недостатком считается необходимость в специальном USB-донгле, в котором находится приемник мышки. Такой донгл занимает USB-слот на компьютере. Потеря донгла делает мышку недееспособной из-за несовместимости методов радиосвязи разных производителей.

Третье поколение радиомышек использует стандартные радиоинтерфейсы. Современные компьютеры оснащаются этим интерфейсом, поэтому мышки с таким интерфейсом не нуждаются в донгле. Другое достоинство — не требуется специальных драйверов. Недостаток — высокая цена и большее энергопотребление.

Индукционные мыши.

Индукционные мыши чаще всего имеют индукционное питание от специальной рабочей площадки («коврика») или графического планшета. Но такие мыши являются беспроводными лишь отчасти — планшет или площадка всё равно подключаются кабелем. Таким образом, кабель не мешает двигать мышью, но и не позволяет работать на расстоянии от компьютера, как с обычной беспроводной мышью.

Достоинства индукционной мыши:

- Очень низкая цена (по сравнению с остальными устройствами наподобие сенсорных экранов);
- Пригодность для длительной работы.
- Высокая точность позиционирования курсора.
- Возможность осуществления разных манипуляций — двойные и тройные щелчки, перетаскивание, нажатие одной кнопки во время перетаскивания другой и т. д. Поэтому в одной руке можно сконцентрировать большое количество органов управления — многокнопочные мыши позволяют осуществлять управление, вообще без привлечения клавиатуры.

Недостатки индукционной мыши:

- Для работы требуется ровная гладкая поверхность достаточных размеров;
- Неустойчивость к вибрациям. По этой причине мышь практически не применяется в военных устройствах.

Трекбол (trackball) функционально представляет собой перевернутую механическую (шариковую) мышь. Шар находится сверху или сбоку, и пользователь может вращать его ладонью или пальцами, не перемещая корпус устройства.

Несмотря на внешние различия, трекбол и мышь конструктивно похожи — при движении шар приводит во вращение пару валиков или, в более современном варианте, его сканируют оптические датчики перемещения (как в оптической мыши).

В настоящее время трекболы достаточно редко применяются в домашних и офисных компьютерах, однако нашли применение в промышленных и военных компьютерах, аппаратах ультразвуковой диагностики, где пользователю приходится работать в условиях недостатка места и возможной вибрации

Джойстик (joystick — «ручка управления самолётом») представляет собой качающуюся в двух плоскостях ручку. Наклоняя ручку вперёд, назад, влево и вправо, пользователь может передвигать что-либо по экрану.

На ручке, а также в платформе, на которой она крепится, обычно располагаются кнопки и переключатели различного назначения. Помимо координатных осей X и Y,

возможно также изменение координаты Z, за счет вращения рукояти вокруг оси, наличия второй ручки, дополнительного колёсика и т. п.

Основным предназначением джойстиков является управление в играх требующих точное позиционирование элемента управления – авиа- и космических симуляторах.

Сенсорная панель (или **тачпад** - touchpad) - это устройство ввода, применяемое в ноутбуках, служит для перемещения курсора в зависимости от движений пальца пользователя. Используется в качестве замены компьютерной мыши и выполняют те же функции что колеса с трекболами, но не имеют движущихся частей.

Работа сенсорной панели основана на измерении емкости пальца или измерении емкости между сенсорами. Емкостные сенсоры расположены вдоль вертикальной и горизонтальной осей панели, что позволяет определять положение пальца с нужной точностью.

Поскольку работа устройства основана на измерении емкости, оно не будет работать, если водить по нему каким-либо непроводящим предметом, например, основанием карандаша. В случае использования проводящих предметов сенсорная панель будет работать только при достаточной площади соприкосновения, поэтому, например, работа с влажными пальцами весьма затруднена.

Преимуществами сенсорных панелей являются:

- отсутствие необходимости в ровной поверхности, как для мыши;
- расположение сенсорной панели, как правило, фиксировано относительно клавиатуры;
- для перемещения курсора на весь экран достаточно лишь небольшого перемещения пальца;
- работа с ними не требует особого привыкания, как, например, в случае с трекболом.

Недостатком же сенсорных панелей является низкое разрешение, что затрудняет работу в графических редакторах и 3D-играх.

Пенмаус - представляет собой аналог шариковой авторучки, на конце которой вместо пишущего узла установлен узел, регистрирующий величину перемещения.

Световое перо — один из инструментов ввода графических данных в компьютер, разновидность манипуляторов. Внешне имеет вид шариковой ручки или карандаша. Обычно на световом пере имеется одна или несколько кнопок, которые могут нажиматься рукой, удерживающей перо. Ввод данных с помощью светового пера заключается в прикосновениях или проведении линий пером по поверхности экрана монитора. В наконечнике пера устанавливается фотоэлемент, который регистрирует изменение яркости экрана в точке, с которой соприкасается перо, за счёт чего соответствующее программное обеспечение вычисляет позицию, «указываемую» пером на экране и может, в зависимости от необходимости, интерпретировать её тем или иным образом, обычно как указание на отображаемый на экране объект или как команду рисования. Кнопки используются аналогично кнопкам мыши — для выполнения дополнительных операций и включения дополнительных режимов.

Световое перо невозможно использовать с обычными ЖК мониторами. Для этого нужен специальный экран, который может реагировать на световое перо.

Средства речевого ввода позволяют пользователю вместо клавиатуры, мыши и других устройств использовать речевые команды (или проговаривать текст, который должен быть занесен в память в виде текстового файла). Возможности таких устройств достаточно ограничены, хотя они постоянно совершенствуются. Проблема не в том, чтобы записать речь, подвергнуть ее дискретизации и ввести коды в компьютер (при современном уровне техники это несложно), а чтобы распознать смысл речи и представить ее, например, в текстовой форме, допускающей последующую компьютерную обработку.

Многие специалисты связывают с прогрессом устройств речевого ввода будущее компьютерной техники, считая такие устройства ведущими элементами ее интеллектуализации.

10.4.2. Устройства вывода

10.4.2.1. Устройства вывода на бумажные носители

Принтер (print - печатать) – устройство для вывода на печать текстовой и графической информации.

Классификация устройств

- 1) В зависимости от **порядка вывода информации** на носитель записи различают *посимвольные*, *построчные* и *постраничные* принтеры.

Посимвольные принтеры выводят алфавитно-цифровую информацию на носитель записи последовательно символ за символом, при этом за один цикл печати формируется один знак.

Построчные принтеры формируют и выводят за один цикл печати всю строку, делая ее доступной для восприятия пользователем сразу же после завершения процесса.

Постраничные принтеры формируют и выводят за один цикл печати целиком страницу.

- 2) По **принципу формирования изображений** символов на носителе записи различают *литерные* (полносимвольные) и *матричные* (знакосинтезирующие) принтеры.

В **литерных** принтерах изображение формируют одновременно по всей поверхности символа при единичном воздействии на носитель записи печатающей головки, например, молоточка.

В **матричных** принтерах изображение символов формируют из отдельных элементов - точек последовательно или последовательно-параллельно при многократном воздействии на носитель записи.

- 3) По **физическому принципу** (технологии печати) - различают принтеры *ударного* и *безударного* действия.

В принтерах **ударного действия** изображение символьной и графической информации получают в результате механического воздействия (удара) печатающего элемента (печатающего знака, молоточка, стержня) на бумагу, как правило, через красящую ленту, из которой выдавливается краситель. В ряде устройств используется прямая печать, при которой краситель (краска) наносится (накатывается) непосредственно на поверхность литеры и далее при ударе переносится на бумагу. Принтеры ударного принципа действия разработаны двух классов - посимвольные и построчные.

В принтерах **безударного действия** изображение на носителе записи получают в результате физико-химического или другого вида воздействия на конечный носитель записи, поступающий к пользователю, или на некоторый промежуточный носитель записи, входящий, как правило, неотъемлемой частью в состав принтера. С промежуточного носителя записи или с его помощью информацию переносят на окончательный носитель записи.

Печатающие устройства ударного действия

Наиболее распространенным типом принтеров ударного действия являются **матричные** принтеры. Принцип печати этих устройств таков: печатающая головка принтера содержит вертикальный ряд тонких металлических стержней (иглоков). Головка движется вдоль печатаемой строки, а стержни в нужный момент ударяют по бумаге через красящую ленту, заправленную в специальную кассету (картридж). Это и обеспечивает формирование на бумаге символов и изображений. Переход к следующей строке достигается синхронизированным движением бумаги (рис.47.).

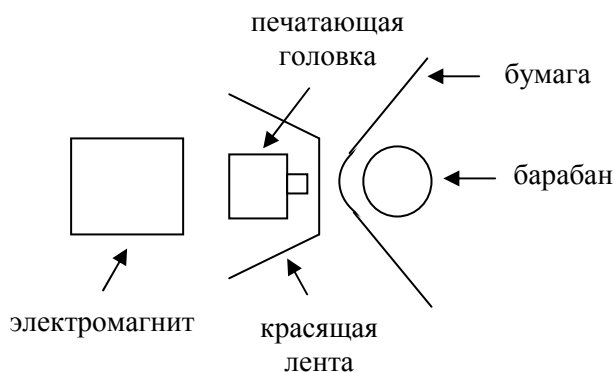


Рис. 47. Принцип действия матричного принтера

Такой подход к формированию символа на носителе записи определяет практически неограниченное разнообразие (номенклатуру) и число наборов символов, печатаемых принтером, возможность вывода не только алфавитно-цифровой, но и графической информации, возможность вывода многоцветных и полутоновых изображений. При этом качество получаемого изображения благодаря возможности повышения плотности печати во многих случаях не уступает полиграфическому.

Современные игольчатые принтеры используют печатающую головку с 9 или 24 иглами, управляемыми при помощи магнитов. Быстродействие последних и количество печатающих игл в основном определяют скорость печати.

Быстродействие данных принтеров при печати простейшими шрифтами, особенно 24-игольчатых, очень высоко и достигает нескольких десятков листов, в минуту.

Игольчатые принтеры имеют гибкие возможности вывода других шрифтов программно с применением соответствующих драйверов и различных форматов матрицы символа, с управлением межсимвольным и междустрочным расстоянием.

Современные принтеры данной группы предусматривают работу с форматами бумаги А4 или А3, различные способы подачи бумаги, печатают на прямом и обратном ходе каретки, имеют удобный пользовательский интерфейс. Эти принтеры очень надежны и имеют низкие эксплуатационные расходы.

Матричные принтеры, несмотря на то, что многие считают их устаревшими, все еще активно используются для печати в лабораториях, банках, бухгалтериях, в библиотеках для печати на карточках, для печати на многослойных бланках (например, на авиабилетах), а также в тех случаях, когда необходимо получить второй экземпляр документа через копирку (обе копии подписываются через копирку одной подписью для предотвращения внесения несанкционированных изменений в финансовый документ).

Принтеры стали “интеллектуальными”, т.е. имеют собственное ОЗУ и электронный блок управления для того, чтобы разгрузить основное ОЗУ и не отнимать в процессе печати время у центрального процессора.

Существуют ударные точечно-матричные принтеры цветной печати. В них используются 4-цветные ленты, и каждая точка изображения формируется четырьмя последовательными ударами иголки разной силы. Таким образом, можно сформировать на бумаге точки всех основных цветов и множества оттенков.

Печатающие устройства безударного принципа действия

Применяются три класса принтеров безударного принципа действия - посимвольные, строчные и постраничные. Во всех классах таких принтеров изображения символьной и графической информации синтезируют из отдельных точек - элементов изображения.

Наибольшее распространение при реализации безударных принтеров получили следующие способы формирования изображения:

- струйный,
- электрографический (электрофотографический),
- магнитографический (феррографический),
- электростатический,
- термический.

Струйные принтеры

Принцип действия струйных принтеров похож на матричные принтеры тем, что изображение на носителе формируется из точек. Но вместо головок с иглками в струйных принтерах используется матрица, печатающая жидкими красителями.

В струйных принтерах краска под давлением выбрасывается из отверстий (сопел) в печатающей головке и затем прилипает к бумаге. При этом формирование изображения происходит как бы из отдельных точек - "клякс", таким образом, соответствующая матрица печати представляет собой набор сопел, с которыми соединены емкости для чернил и управляющие механизмы (рис. 48).

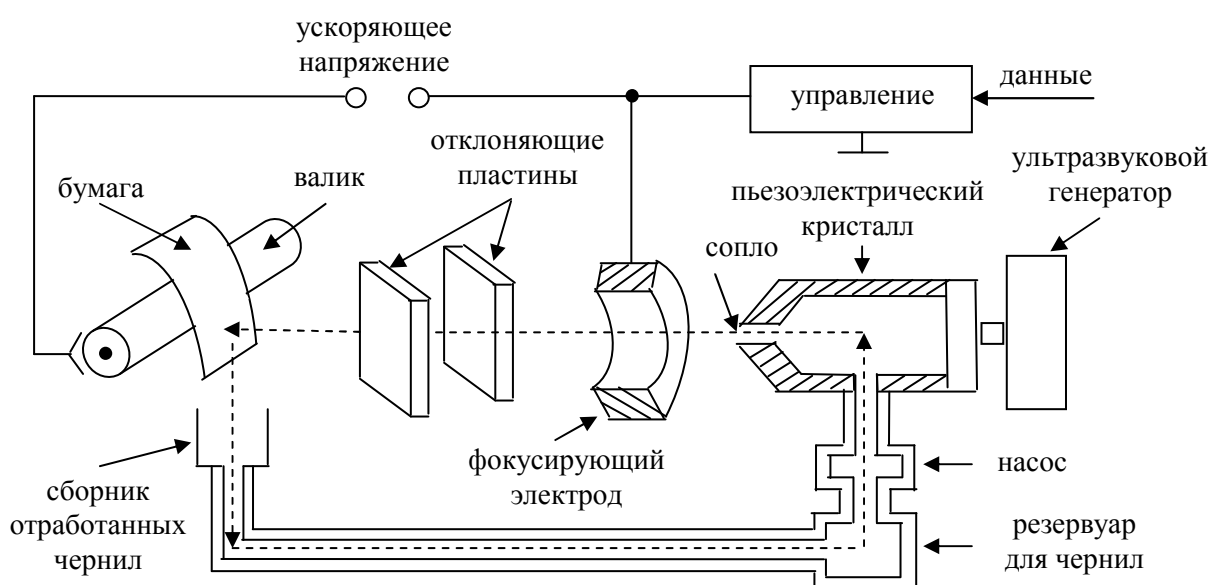


Рис.48. Принцип действия струйного принтера

Струйные принтеры нашли очень широкое применение в цветной печати, поскольку струйный способ позволяет реализовывать не только одноцветную, но и многоцветную печать. Для этого в блоке головок располагают, как правило, четыре группы сопел, каждая из которых связана с емкостью, заполненной чернилами одного из четырех цветов: черного, синего, пурпурного и желтого, что позволяет получать 7-цветные изображения. При тонкой дозировке и формировании одного элемента изображения из большого количества микрокапель получают изображения, состоящие из еще большего числа цветов.

Достоинства струйного способа печати:

- более высокое качество печати по сравнению с матричными принтерами,
- относительно небольшое количество движущихся механических частей и, соответственно, простоту и надежность механической части устройства,
- меньший уровень шума при работе,
- относительно низкую стоимость.

Недостатки:

- нестабильность получаемого разрешения, что ограничивает возможность их применения в черно-белой полутонной печати,

- необходимость тщательного ухода и обслуживания, поскольку при длительном простое принтера происходит высыхание остатков красителя на соплах печатающей головки,
- противоречивые и высокие требования к краскам,
- сильная зависимость качества изображения от типа бумаги,
- характерна высокая стоимость расходных материалов.

Электрографические и **магнитографические** принтеры. Основное отличие между этими типами устройств состоит в том, что в электрографических принтерах создают скрытое электрическое изображение, а в магнитографических - магнитное.

Электрографические принтеры. Во всех электрографических устройствах создается скрытое электрическое изображение на промежуточном носителе записи с фотопроводниковым слоем на поверхности, далее это изображение визуализируют мелкодисперсным красящим порошком – *тонером*. Получающееся при этом порошковое изображение переносят на окончательный носитель записи - бумагу, где и закрепляют - фиксируют, например, термическим способом.

На фотопроводниковый носитель записи действуют электромагнитным излучением, источниками которого могут быть лазеры, светодиоды, светоклапанные системы или электронно-лучевые трубки. Часто электрографические принтеры различают именно по виду используемого в них источника излучения (лазерные, светодиодные и т.п.).

В настоящее время наилучшее (близкое к типографскому) качество печати достигается в *лазерных* принтерах. В этих принтерах для печати используется принцип ксерографии: изображение переносится на бумагу со специального барабана, к которому электрически притягиваются частички краски. Отличие лазерного принтера от обычного ксерокопировального аппарата состоит в том, что печатающий барабан электризуется с помощью лазера по командам из компьютера.

Лазерные принтеры формируют изображение путем позиционирования точек на бумаге (растровый метод). Первоначально страница формируется в памяти принтера, и лишь затем передается в механизм печати. Растровое представление символов и графических образов производится под управлением контроллера принтера. Каждый образ формируется путем соответствующего расположения точек в ячейках сетки или матрицы.

В лазерных и светодиодных печатающих устройствах используется свойство фоточувствительности ряда материалов, которые изменяют свой поверхностный электростатический заряд под воздействием света (рис.49.).

Один из основных узлов лазерного принтера – вращающийся фото-барабан, на внешней поверхности которого нанесен специальный светочувствительный материал. Фото-барабан представляет собой металлический цилиндр, покрытый тонкой пленкой из фотопроводящего полупроводника (обычно оксид цинка). По поверхности барабана равномерно распределяется статический заряд.

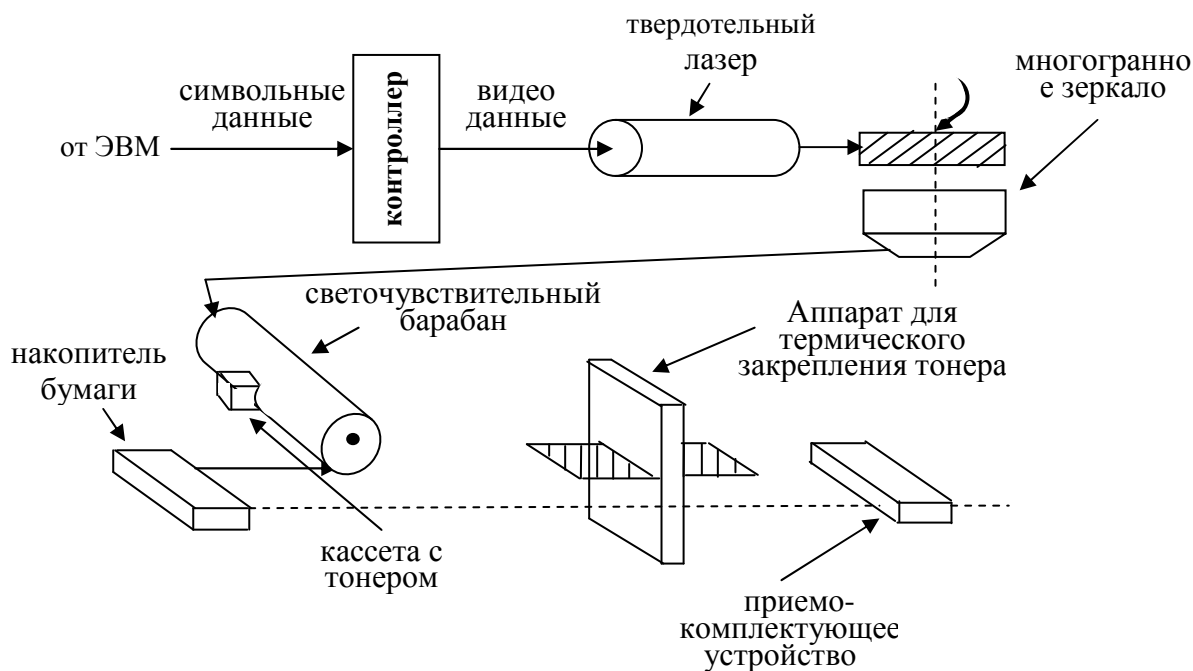


Рис.49. Принцип действия лазерного принтера

1. Управляемый электронным блоком (микроконтроллером) луч лазера, отражаясь от вращающегося зеркала, пробегает по фото-барабану и засвечивает на нем элементарные площадки (точки); в результате фотоэлектрического эффекта в этих точках изменяется электрический заряд.
2. На следующем рабочем шаге с помощью другого барабана, называемого девелопером (developer), на фотобарабан наносится тонер - мельчайшая красящая пыль. Под действием статического заряда мелкие частицы тонера легко притягиваются к поверхности барабана в точках подвергшихся экспозиции, и формируют на нем изображение.
3. Далее лист бумаги из подающего лотка помощью системы валиков перемещается к барабану. Затем листу сообщается статический заряд, противоположный по знаку заряду засвеченных точек на барабане. При соприкосновении бумаги с барабаном частички тонера с барабана переносятся (притягиваются) на бумагу.
4. Для фиксации тонера на бумаге листу вновь сообщается заряд и он пропускается между двумя роликами нагревающими его до температуры около 180' - 200°C.

После собственно процесса печати барабан полностью разряжается, очищается от прилипших частиц тонера и готов для нового цикла печати. Описанная последовательность действий происходит очень быстро и обеспечивает высокое качество печати.

Особенности данного процесса, такие, как формирование точки изображения лучем света, а далее - мелкодисперсным специальным порошком красителя, определяют возможность очень малых размеров точки матрицы изображения и соответственно - разрешающую способность лазерных принтеров, которая на практике составляет 300-1200 точек на дюйм.

В целом высокая разрешающая способность принтеров данной группы позволяет использовать их для печати разнообразной текстовой и графической информации, вплоть до изготовления полиграфических макетов и форм.

Для обеспечения печати графики лазерные устройства, как правило, имеют буферную память.

Достоинства лазерных принтеров:

1. Лазерные принтеры - рекордсмены по части количества воспроизводимых шрифтов и качеству рисунков благодаря высочайшей разрешающей способности.
2. Существуют как черно-белые, так и цветные лазерные принтеры.
3. Лазерный принтер работает почти бесшумно.
4. Высокая скорость печати (печатают быстрее струйных и др. принтеров).
5. Относительно небольшое время, необходимое для приведения оборудования в состояние готовности.
6. Возможность использования разной (например, текстурной) бумаги и плёнки.
7. Отпечатки с лазерного принтера более стойки к влаге, агрессивным средам.

Недостатки:

1. Поскольку тонер термически напекается на носитель, со временем может происходить осыпание изображения, особенно если бумага подвергается механическому воздействию.
2. Для лазерных принтеров краситель (тонер) является не единственным расходным материалом. Регулярной замены требует фото-барабан.

Светодиодные принтеры. Принцип действия светодиодных принтеров похож на принцип действия лазерных принтеров. Разница заключается в том, что источником света является не лазерная головка, а линейка светодиодов.

Электростатические принтеры. Электростатическая регистрация состоит в создании скрытого электрического изображения на специальной диэлектрической поверхности оконечного или промежуточного носителя. Наибольшее распространение получили электростатические устройства без промежуточного носителя. В них запись ведется на специальную электростатическую бумагу, рабочая поверхность которой имеет тонкий диэлектрический слой. Основная область применения таких устройств - вывод графической информации, включая схемную и конструкторскую документацию.

Феррографические принтеры. Структурные схемы феррографических устройств аналогичны схемам электрографических устройств. Основное отличие между этими типами устройств состоит в том, что в феррографических принтерах создают скрытое магнитное, а не электрическое изображение.

Термические принтеры. Эти принтеры очень близки по механизму к матричным (они используют печатную головку, оснащенную матрицей нагревательных элементов, и специальную бумагу, пропитанную термочувствительным красителем). Способ термопечати основан на воздействии на носитель теплотой, выделяемой записывающей (термопечатающей) головкой, которая содержит от нескольких единиц до нескольких тысяч отдельных элементов.

Все большее распространение находят термопечатающие устройства, использующие промежуточный носитель. По этой схеме между термопечатающей головкой и оконечным носителем, в качестве которого используют бумагу, пластмассовую пленку и т.п., размещают промежуточный носитель - копировальную пленку, которая представляет собой тонкую пластмассовую пленку, покрытую со стороны, обращенной к оконечному носителю, красящим слоем. Особенностью этого слоя является низкая (менее 100° С) температура плавления связующего вещества красящего слоя, в котором распределен краситель.

При контакте оконечного носителя с красящим слоем промежуточного носителя и при кратковременном прогреве его термопечатающей головкой красящий слой локально оплавляется и практически полностью переходит на оконечный носитель, создавая на нем элемент изображения. Его цвет определяется цветом красящего вещества. При

последовательном переносе элементов на один и тот же конечный носитель с промежуточных носителей разных цветов получают многоцветные изображения. Допускается наложение одних отпечатков на другие, что расширяет цветовую гамму изображения

Достоинствами термопринтеров являются малый уровень шума при работе, компактность, надежность, отсутствие заправляемых расходных материалов.

Недостатками такого принтера является часто недостаточная яркость и контрастность изображения, номенклатура доступных типов бумаги.

Замечание. Существуют принтеры, работающие на других физических принципах, но по распространенности в настоящее время они значительно уступают тем, которые обсуждались выше.

Принцип работы 3D принтера

Трехмерный или 3D принтер - это устройство вывода трехмерных данных (как правило, объемной геометрии). То есть результатом его работы является некоторый физический объект.

Существует несколько технологий объемной (трехмерной, 3D) печати, но в основе любой из них лежит принцип послойного создания твердой геометрии.

Принцип работы одного из типов 3D принтеров более всего схож с работой обычного струйного принтера. Основное отличие заключается в том, что вместо нанесения чернил из печатающей головки на очередной лист бумаги в принтере связующее вещество через печатающую головку наносится на очередной тонкий (около 0,1 мм) слой порошка, создавая одно сечение объекта. В тех местах, где было нанесено связующее вещество, порошок твердеет. Следующее сечение "склеивается" с предыдущим и так далее, пока не будет сформировано полностью твердое тело. После окончания работы 3D принтера изделие извлекается из массы порошка.

Плоттер или **графопостроитель** (plotter, от to plot — вычерчивать чертежи),— это устройство вывода графической информации из ЭВМ на бумажный носитель.

Эти устройства нужны для вывода схем, сложных архитектурных чертежей, художественной и иллюстративной графики, карт, трехмерных изображений. Они часто используются для производства высококачественной цветной документации и являются незаменимыми для художников, дизайнеров, оформителей, инженеров, проектировщиков.

Плоттеры являются устройством вывода, которое применяется только в специальных областях. Они обычно используются совместно с программами САПР (Системы Автоматизированного Проектирования). Результат работы практически любой такой программы - это комплект конструкторской и/или технологической документации, в которой значительную часть составляют графические материалы.

Размеры выходных документов на плоттере гораздо больше документов, которые можно вывести из ПК, используя принтер. Максимальная длина печатаемого материала, как правило, зависит от длины рулона бумаги, а не от размеров и конструкции самого плоттера.

Так же как и у принтеров, изображение на бумаге формируется при помощи печатающей головки. Точка за точкой изображение наносится на бумагу (кальку, пленку).

Большинство плоттеров имеют пишущий узел перьевого типа. Используются специальные фломастеры с возможностью их автоматической замена (по сигналу программы) из доступного набора. Кроме фломастеров, применяются чернильные, шариковые пишущие узлы и т.д.

Плоттеры подключаются к компьютеру через параллельный или последовательный интерфейс, либо в слот расширения встраивается плата.

Классификация плоттеров

1. По принципу формирования изображения:

- Плоттеры **векторного типа** имеют пишущий узел, способный перемещаться относительно бумаги сразу по двум направлениям (вертикали и

горизонтали), и изображение на бумаге формируется непосредственно вычерчиванием нужных прямых и кривых в любых направлениях.

- Плоттеры **растрового типа** имеют пишущий узел, который перемещается относительно бумаги только в одном направлении, и изображение формируется строка за строкой из последовательно наносимых точек.

2. По конструкции все современные плоттеры можно отнести к двум большим классам:

- **Планшетные** для формата А3-А2 (реже А1-А0) с фиксацией листа электрическим, реже магнитным или механическим способом, и пишущим узлом. Таким образом, если, например, необходимо провести линию, то печатающий узел перемещается в её начальную точку, опускается штифт с пером, соответствующим толщине и цвету проводимой линии, и затем перо перемещается до конечной точки линии.
- **Барабанные** (рулонные) плоттеры с шириной бумаги формата А1 или А0, роликовой подачей листа, механическим и/или вакуумным прижимом и с пишущим узлом.

Барабанные плоттеры используют рулоны бумаги длиной до нескольких десятков метров и позволяют создавать длинные рисунки и чертежи.

3. По принципу действия:

- **Перьевые** плоттеры используют для получения изображения обычные перья. Для получения цветного изображения используется несколько перьев различного цвета.
- **Струйные** плоттеры формируют изображение подобно струйным принтерам, разбрызгивая капли чернил на бумагу. Качество печати, превосходящее возможности перьевых плоттеров, определяет широкое распространение струйных плоттеров в различных областях человеческой деятельности, включая автоматическое проектирование, инженерный дизайн.
- **Электростатические** плоттеры создают изображение с помощью электрического заряда. Электростатические плоттеры очень дороги и используются, когда требуется высокое качество выходных документов.
- Плоттеры с **термопереносом** создают двухцветное изображение, используя теплочувствительную бумагу и электрически нагреваемые иглы.
- **Карандашные** плоттеры используют для получения изображения обычный грифель. Они самые дешевые и требуют дешевого расходного материала.

К основным характеристикам плоттеров относятся:

- скорость вычерчивания изображения, измеряемая в миллиметрах в секунду;
- скорость вывода, определяемая количеством листов, распечатываемых в минуту;
- разрешающая способность, измеряемая, аналогично принтеру.

10.4.2.2. Устройства вывода на электронные носители

Мониторы

Прежде всего, определим два понятия, которые часто и неточно используют как синонимы – **дисплей** и **монитор**.

Дисплей (**display** — показывать, визуализировать) — электронное устройство, предназначенное для визуального отображения информации. Дисплеем в большинстве случаев можно назвать **часть законченного устройства**, используемую для отображения цифровой, цифро-буквенной или графической информации электронным способом. Понятие дисплея можно ассоциировать с понятием экрана.

Монитор – это конструктивно законченное устройство, предназначенное для визуального отображения информации, и может иметь дисплеи разных типов. Современный монитор состоит из экрана (дисплея), блока питания, плат управления и корпуса.

Монитор с клавиатурой и мышью на данный момент является основным средством оперативного взаимодействия пользователя с ЭВМ и предназначен для вывода на экран текстовой и графической информации. Он позволяет вводить в ЭВМ данные, осуществлять диалог с ЭВМ, редактировать и обрабатывать текстовую и графическую информацию, отображать на экране результаты решения задач в виде таблиц, текста, графиков, рисунков.

Классификацию мониторов можно провести по следующим признакам:

- по физическому принципу формирования изображения на экране,
- по способу построения изображения (виду развертки),
- по длительности хранения информации на экране,
- по цветности изображения,

По принципу формирования изображения. По физическим принципам, лежащим в основе конструкций дисплеев, подавляющее большинство их относится к дисплеям на базе электронно-лучевых трубок (ЭЛТ или **CRT** - от Cathode Ray Tube) и к жидкокристаллическим (ЖК или **LCD** - от Liquid Crystal Display).

У ЭЛТ формирование изображения производится на внутренней поверхности экрана, покрытого слоем люминофора – вещества, светящегося под воздействием электронного луча, генерируемого специальной “электронной пушкой” и управляемого системами горизонтальной и вертикальной развертки.

У ЖК вместо электронного луча, возбуждающего люминофор, для воздействия на жидкие кристаллы, помещенные между двумя слоями стекла, используется электрический заряд. Жидкокристаллический экран состоит из крошечных сегментов, заполненных специальным веществом, способным менять отражательную способность под воздействием очень слабого электрического поля, создаваемого электродами, подходящими к каждому сегменту.

По способу построения изображения (для ЭЛТ) различаются **матричная** и **векторная** развертки.

При **матричной развертке** электронный луч на экране перемещается скачками, высвечивая, таким образом, матрицу точек - **пиксел**. При такой развертке легко перевести луч в любую заданную точку экрана - надо только в счетчики строк и кадров поместить координаты этой точки.

Векторная развертка используется для рисования сложных фигур с помощью сплошных линий разной формы. Управление вертикальным и горизонтальным отклонением луча в этом случае осуществляется с помощью функциональных генераторов, каждый из которых настроен на прорисовку определенного графического примитива. Состав графических примитивов, из которых строится изображение, определяется наличием функциональных генераторов.

По длительности хранения информации на экране мониторы делятся на **регенерируемые** и **запоминающие**.

В **регенерируемых мониторах** изображение после однократной прорисовки держится на экране недолго, доли секунды, постепенно угасая. Угасание изображения иногда заметно на глаз - нижние строки, например, могут быть ярче верхних. Для поддержания постоянной яркости изображение приходится повторно прорисовывать (регенерировать) изображение 20-25 раз в секунду. А чтобы яркость в различных частях экрана не очень отличалась применяют чересстрочную развертку: при каждой прорисовке сначала рисуются нечетные строки, а затем - четные.

Регенерируемые мониторы незаменимы при визуализации быстропротекающих динамических процессов.

В *запоминающих мониторах* после однократной прорисовки изображение держится на экране в течение нескольких часов. Для его стирания приходится подавать на экран специальное стирающее напряжение.

Запоминающие мониторы эффективны там, где выведенное изображение нуждается в длительной обработке, например, подвергается редактированию или должно быть воспринято (изучено).

По цветности изображения различают *монохромные* (черно-белые) и *цветные* мониторы.

CRT-мониторы

Чтобы понять принципы работы растровых и векторных дисплеев с регенерацией, нужно иметь представление о конструкции CRT и методах создания видеоизображения.

На рис.50. показана общая схема устройства CRT.

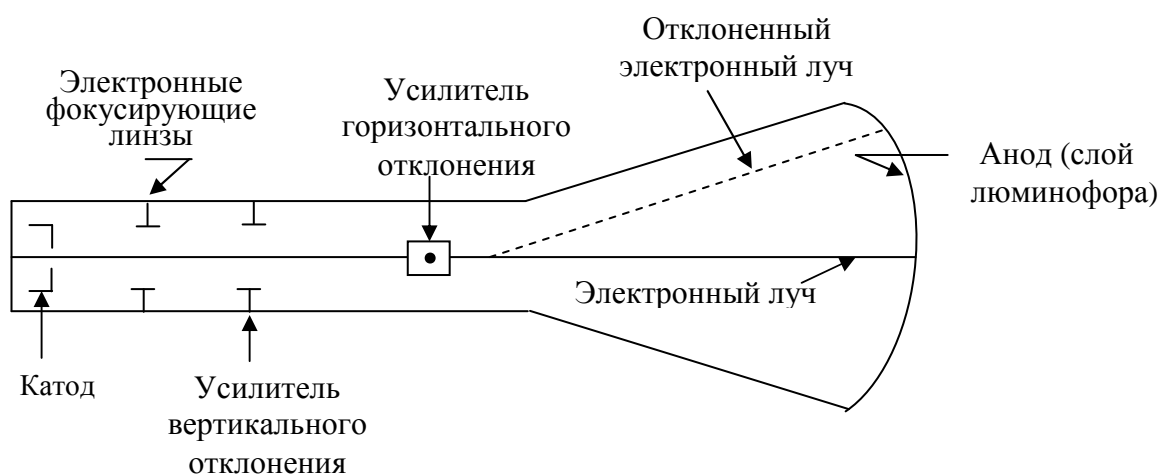


Рис.50. Схема ЭЛТ

Кинескоп состоит из герметичной стеклянной трубки, внутри которой находится вакуум, то есть весь воздух удален. Один из концов трубки узкий и длинный - это горловина, а другой - широкий и достаточно плоский - это экран.

Катод (отрицательно заряженный) нагревают до тех пор, пока возбужденные электроны не создадут расширяющегося облака (электроны отталкиваются друг от друга, так как имеют одинаковый заряд). Эти электроны притягиваются к сильно заряженному положительному аноду. На внутреннюю сторону расширенного конца "КН нанесен люминофор, вещество, которое испускает свет при бомбардировке его заряженными частицами. Электроны попадают на люминофорный слой, после чего энергия электронов преобразуется в свет, т.е. поток электронов заставляет точки люминофора светиться. Эти светящиеся точки люминофора и формируют изображение.

Облако электронов с помощью линз фокусируется в узкий, строго параллельный пучок (электронная пушка), и луч дает яркое пятно в центре ЭЛТ. Луч отклоняется (позиционируется) влево или вправо от центра и (или) выше или ниже центра с помощью усилителей горизонтального и вертикального отклонения (отклоняющая система). Именно в этот момент проявляется отличие *векторных* и *растровых* дисплеев.

В отличие от этого в **растровом** дисплее луч может отклоняться только в строго определенные позиции на экране, образующие своеобразную мозаику. Эта мозаика составляет видеоизображение. Люминофорное покрытие на экране растровой ЭЛТ тоже не непрерывно, а представляет собой множество тесно расположенных мельчайших точек, куда может позиционироваться луч, образуя мозаику.

Как правило, в цветном CRT-мониторе используется три электронные пушки (RGB), в отличие от одной пушки, применяемой в монохромных мониторах, которые сейчас практически не производятся.

Для управления электронно-лучевой трубкой необходима и управляющая электроника, качество которой во многом определяет и качество монитора. Кстати, именно различие в качестве управляющей электроники, создаваемой разными производителями, является одним из критериев определяющих разницу между мониторами с одинаковой электронно-лучевой трубкой.

С электронно-лучевой трубкой дисплея связан видеопроцессор, в состав которого входит **буферная память**, несущая всю необходимую информацию для управления движением луча (экранная память, память регенерации, видеопамять, видеобуфер). В частности, в видеопамяти размещаются данные (коды символов) для отображения на экране. Размер и расположение буферов меняется с видеосистемой, режимом экрана, количеством заранее отведенной памяти

Векторные дисплеи

В **векторном** дисплее электронный луч может быть отклонен непосредственно из любой произвольной позиции в любую другую произвольную позицию на экране ЭЛТ. Поскольку люминофорное покрытие нанесено на экран ЭЛТ сплошным слоем, в результате получается почти идеальная прямая. Таким образом, изображение на экране задается непрерывным движением электронного луча по заданной траектории. Траектория движения определяется ломаной линией, состоящей из векторов, задаваемых координатами начальной и конечной точки. Эти координаты можно задавать хотя и с ограниченной, но с достаточно высокой точностью. Тем самым на экране создается видимость гладких непрерывных линий. В векторных дисплеях можно управлять не только движением, но и интенсивностью свечения, получая эффект тонких и толстых линий рисунка (рис.51.).

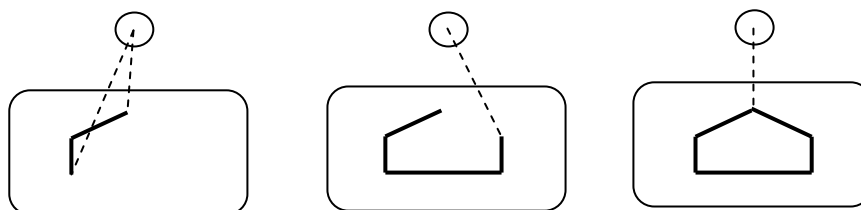


Рис.51. Векторный режим построения изображения

Кроме ЭЛТ, для векторного дисплея необходим **дисплейный буфер** и **дисплейный контроллер**.

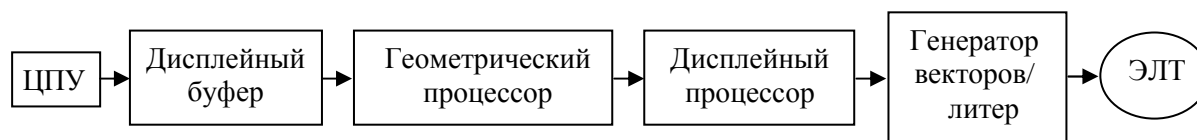
Дисплейный буфер - непрерывный участок памяти, содержащий всю информацию, необходимую для вывода изображения на ЭЛТ.

Функция **дисплейного контроллера** заключается в том, чтобы циклически обрабатывать информацию в дисплейном буфере со скоростью регенерации. Сложность рисунка ограничивается двумя факторами - размером дисплейного буфера и скоростью контроллера.

Видеопамять векторных дисплеев содержит последовательность координат векторов, определяющих движение луча по всему экрану, и указания об его интенсивности на том или ином участке траектории движения. Каждый знак (литера) на таком экране задается также последовательностью векторов.

Функции внутреннего программного обеспечения векторных дисплеев состоят в том, чтобы своевременно заполнить экранную память командами, определяющими маршруты электронного луча так, как требует видеоконтроллер, который их и преобразует в сигналы, управляющие электроникой, ведающей отклонением луча. При этом следует отметить, что чем сложнее рисунок (чем больше векторов и перемещений в нем задано), тем больше времени затрачивает дисплейный процессор на опрос команд буферной памяти и их анализ. Сложные рисунки приводят в силу этого к эффекту мерцания, так как темп повторения рисунка (темп регенерации) снижается. Одной из технических характеристик векторных дисплеев является допустимое число векторов, которое можно задавать до появления мерцания на экране.

На рис.52. изображена одна из возможных блок-схем высокопроизводительного векторного дисплея. Предполагается, что такие геометрические преобразования, как поворот, перенос, масштабирование, перспективное проецирование и отсечение, реализованы аппаратно в *геометрическом процессоре*.



ЦПУ – центральное процессорное управление.

Рис.52. Блок-схема векторного дисплея с регенерацией

Геометрический процессор работает быстрее, чем необходимо для регенерации достаточно сложных изображений. В этом случае исходная геометрическая база данных, посланная из ЦПУ, сохраняется непосредственно в дисплейном буфере, а векторы обычно задаются в пользовательских координатах в виде чисел с плавающей точкой.

Дисплейный контроллер за один цикл регенерации считывает информацию из дисплейного буфера, пропускает ее через геометрический процессор и результат передает генератору векторов. При таком способе обработки геометрические преобразования должны выполняться "на лету" в течение одного цикла регенерации.

Видеопамять векторных дисплеев содержит последовательность координат векторов, определяющих движение луча по всему экрану, и указания об его интенсивности на том или ином участке траектории движения. Каждый знак на таком экране задается также последовательностью векторов.

Функции внутреннего программного обеспечения векторных дисплеев состоят в том, чтобы своевременно заполнить экранную память командами, определяющими маршруты электронного луча так, как требует видеоконтроллер, который их и преобразует в сигналы, управляющие электроникой, ведающей отклонением луча. При этом следует отметить, что чем сложнее рисунок (чем больше векторов и перемещений в нем задано), тем больше времени затрачивает дисплейный процессор на опрос команд буферной памяти и их анализ. Сложные рисунки приводят в силу этого к эффекту мерцания, так как темп повторения рисунка (темп регенерации) снижается. Одной из технических характеристик векторных дисплеев является допустимое число векторов, которое можно задавать до появления мерцания на экране.

Растровые дисплеи

В отличие от векторных, в **растровом** дисплее луч может отклоняться горизонтально только в строго определенные позиции на экране, образующие своеобразную мозаику. Эта мозаика и составляет видеоизображение. Люминофорное покрытие на экране растровой ЭЛТ тоже не непрерывно, а представляет собой множество тесно расположенных мельчайших точек, куда может позиционироваться луч, образуя мозаику.

Путь электронного луча на экране схематично показан на рис.53. Сплошные линии - это прямой ход луча, пунктир - обратный.

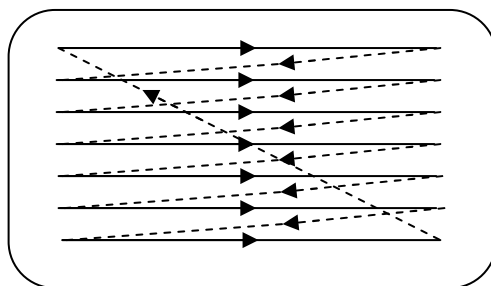


Рис. 53. Ход луча в ЭЛТ-мониторе

Частота перехода на новую линию называется частотой горизонтальной (или строчной) развертки. Частота перехода из нижнего правого угла в левый верхний называется частотой вертикальной (или кадровой) развертки.

Растровые дисплеи предназначены для получения изображений в виде совокупности точек, отрезков векторов, символов и закрашенных областей с помощью матрицы точечных элементов изображений - **пикселей** (от английского *picture element*). Движение луча (или нескольких лучей при построении цветного изображения) происходит построчно. Изображение получается путем изменения интенсивности луча во время сканирования строк. Тем самым свечение возможно только в дискретных точках, которые и образуют точечный растр экрана. Такой принцип аналогичен игольчатым точечным печатающим устройствам (рис.54.).

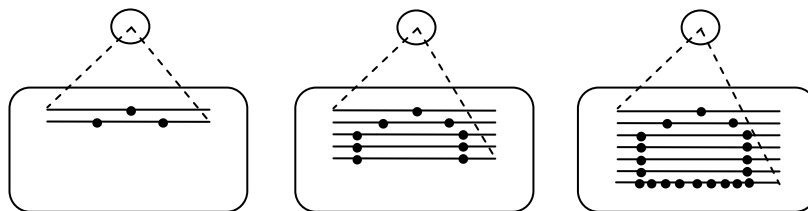


Рис. 54. Растровый способ построения изображений

Невозможно, за исключением специальных случаев, непосредственно нарисовать отрезок прямой из одной адресуемой точки (пиксела) в другую адресуемую точку. Отрезок можно только аппроксимировать последовательностями точек, близко лежащих к реальной траектории отрезка (рис.55.).

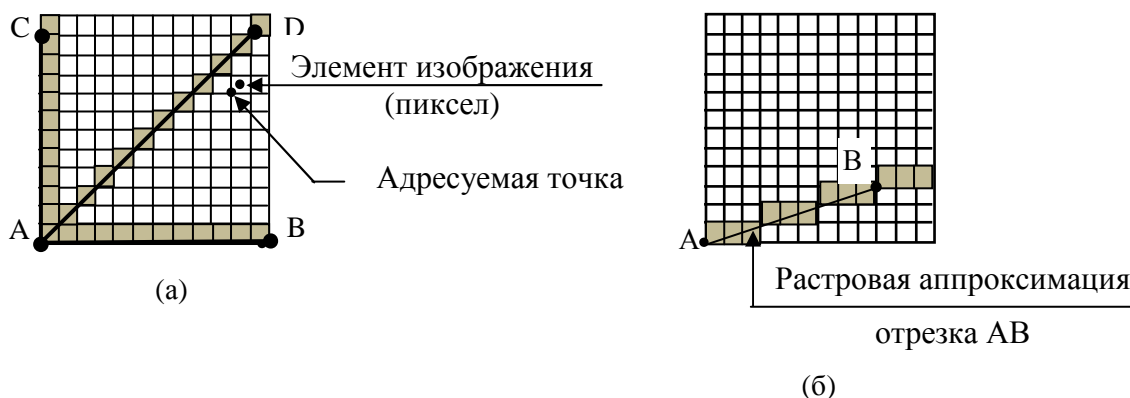


Рис.55. Растровая развертка отрезка

(а) Отрезок прямой из точек получится только в случае горизонтальных, вертикальных или расположенных под углом 45° отрезков .

(б) Все другие отрезки будут выглядеть как последовательности ступенек. Это явление называется лестничным эффектом или "зазубренностью".

Чаще всего для графических устройств с растровой ЭЛТ используется **буфер кадра**. Буфер кадра представляет собой большой непрерывный участок памяти компьютера. Для каждого пиксела в растре отводится как минимум один бит памяти. Эта память называется **битовой плоскостью**. Для квадратного растра размером 512×512 требуется 2^{18} , или 262144 бита памяти в одной битовой плоскости. Из-за того, что бит памяти имеет только два состояния (двоичное 0 или 1), имея одну битовую плоскость, можно получить лишь черно-белое изображение.

Битовая плоскость является цифровым устройством, тогда как растровая ЭЛТ - аналоговое устройство. Поэтому при считывании информации из буфера кадра и ее выводе на графическое устройство необходимо произвести преобразование из цифрового представления в аналоговый сигнал. Такое преобразование выполняет цифро-аналоговый

преобразователь (ЦАП). На рис.56. приведена схема графического устройства с черно-белой растровой ЭЛТ, построенного на основе буфера кадра с одной битовой плоскостью.

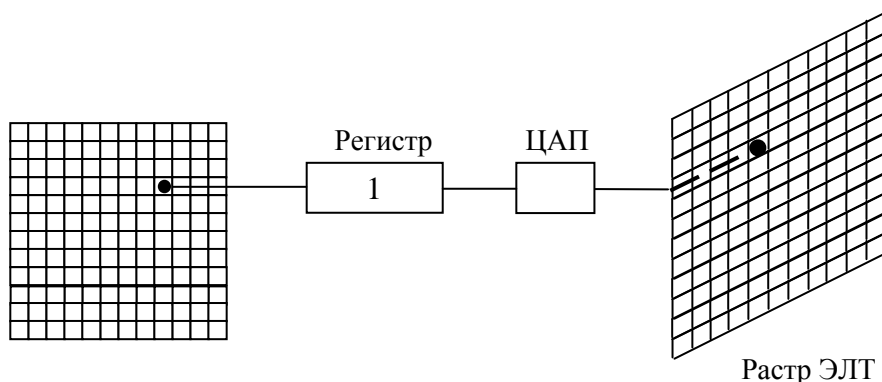


Рис.56. Черно-белый буфер кадра (с одной битовой плоскостью)

Построение цветного изображения

При построении цветного изображения люминофор наносится в виде наборов точек трёх основных цветов — красного, зелёного и синего; любой цвет спектра можно представить их сочетаниями в различных пропорциях (рис 57.).

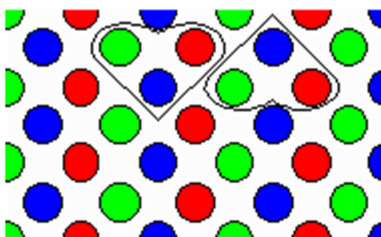


Рис.57. Пиксельные триады

Наборы точек люминофора располагаются по треугольным триадам. Триада образует **пиксел (pixel** — picture element, элемент картинки) — точку, из которых формируется изображение.

Расстояние между центрами пикселов называется **точечным шагом монитора**. Это расстояние существенно влияет на чёткость изображения. Чем меньше шаг, тем выше чёткость. Обычно в цветных мониторах шаг составляет 0,24 мм. При таком шаге глаз человека воспринимает точки триады как одну точку "сложного" цвета.

Схема ЭЛТ для построения цветных изображений представлена на рис.58.

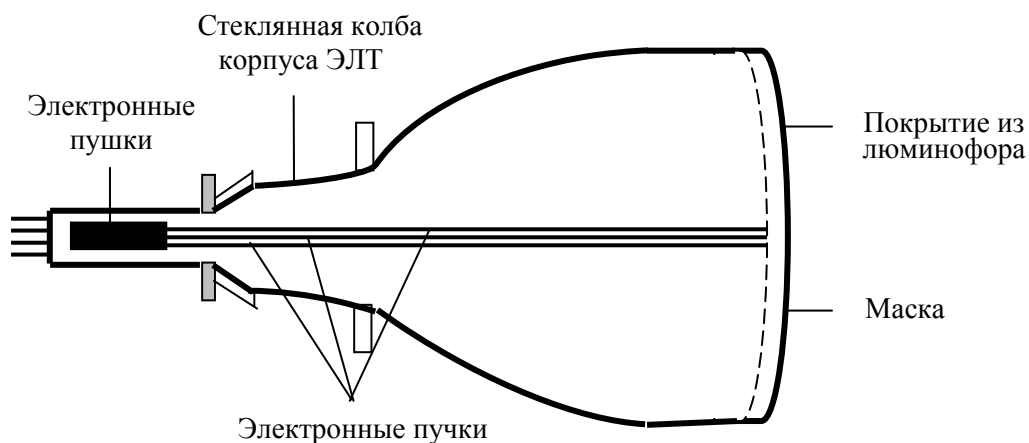


Рис.58. Схема цветной ЭЛТ

На противоположной стороне трубки расположены три (по количеству основных цветов) электронные пушки. Все три пушки "нацелены" на один и тот же пиксел, но каждая из них излучает поток электронов в сторону "своей" точки люминофора. Перед экраном на пути электронов ставится **маска** — тонкая металлическая пластина с большим количеством отверстий, расположенных напротив точек люминофора. Маска обеспечивает попадание электронных лучей только в точки люминофора соответствующего цвета.

Цвета могут быть введены в буфер кадра путем использования дополнительных битовых плоскостей. Поскольку существует три основных цвета, можно реализовать простой цветной буфер кадра с тремя битовыми плоскостями, по одной для каждого из основных цветов. Каждая битовая плоскость управляет индивидуальной электронной пушкой для каждого из трех основных цветов. Три основных цвета, комбинируясь на ЭЛТ, дают восемь цветов. Схема простого цветного растрового буфера показана на рис.59. Чтобы увеличить количество цветов для каждой из трех цветных пушек используются дополнительные битовые плоскости.

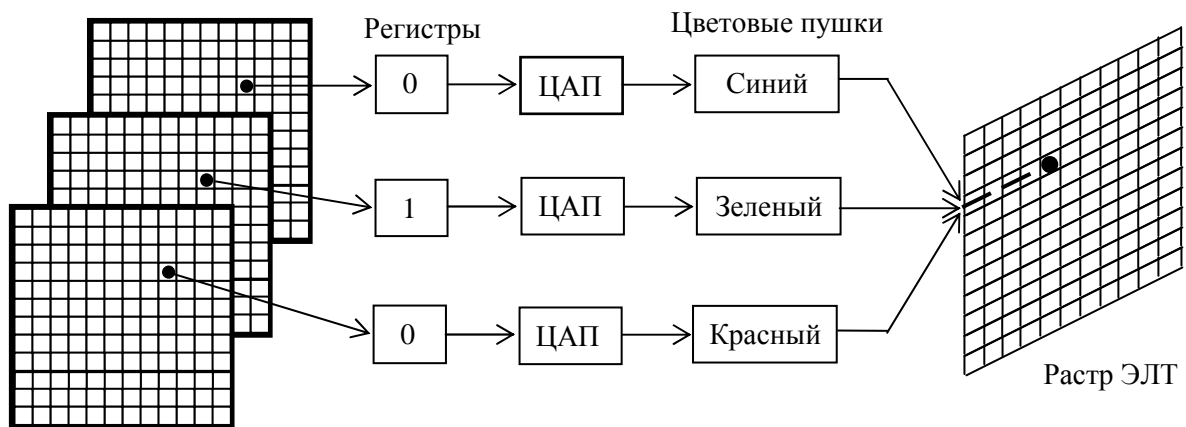


Рис.59. Простой цветной буфер кадра

При выводе на экран любого изображения, независимо от того, в растровом или векторном форматах оно зафиксировано в графических файлах, в видеопамати формируется информация растрового типа. Каждый пиксел однозначно связан с долей видеопамати — несколькими битами, в которых программным путем задаются характеристики пиксела: яркость и, при цветном экране, цветность свечения этого пиксела. Специальная системная программа десятки раз в секунду считывает содержимое видеопамати и обновляет содержимое каждого пиксела, тем самым создавая и поддерживая на экране изображение.

Величиной электронного тока пушек и, следовательно, яркостью свечения пикселов, управляет сигнал, поступающий с видеоадаптера.

На ту часть колбы, где расположены электронные пушки, надевается **отклоняющая система** монитора, которая заставляет электронный пучок пробегать поочередно все пикселы строчку за строчкой от верхней до нижней, затем возвращаться в начало верхней строки и т.д.

Количество отображённых строк в секунду называется **строчной частотой развертки**. А частота, с которой меняются кадры изображения, называется **кадровой частотой развёртки**. Последняя не должна быть ниже 85 Гц, иначе изображение будет мерцать.

LCD-мониторы

LCD-мониторы значительно отличаются от мониторов на электронно-лучевых трубках по принципу работы. Вместо электронного луча, возбуждающего люминофор, в этих дисплеях для воздействия на жидкие кристаллы, помещенные между двумя слоями стекла, используется электрический заряд. В результате получается устройство с существенно другими потребительскими свойствами: малое потребление энергии, отсутствие геометрических искажений, плоская конструкция.

Жидкокристаллический экран состоит из крошечных сегментов, заполненных специальным веществом, способным менять отражательную способность под воздействием очень слабого электрического поля, создаваемого электродами, подходящими к каждому сегменту.

Работа ЖК монитора. Поперечное сечение представляет собой многослойный бутерброд (рис.60.). Крайний слой любой из сторон выполнен из стекла. Между этими слоями расположен тонкопленочный транзистор, панель цветного фильтра, обеспечивающая нужный цвет – красный, синий или зеленый, и слой жидких кристаллов. Вдобавок ко всему существует флуоресцентная подсветка, освещающая экран изнутри.

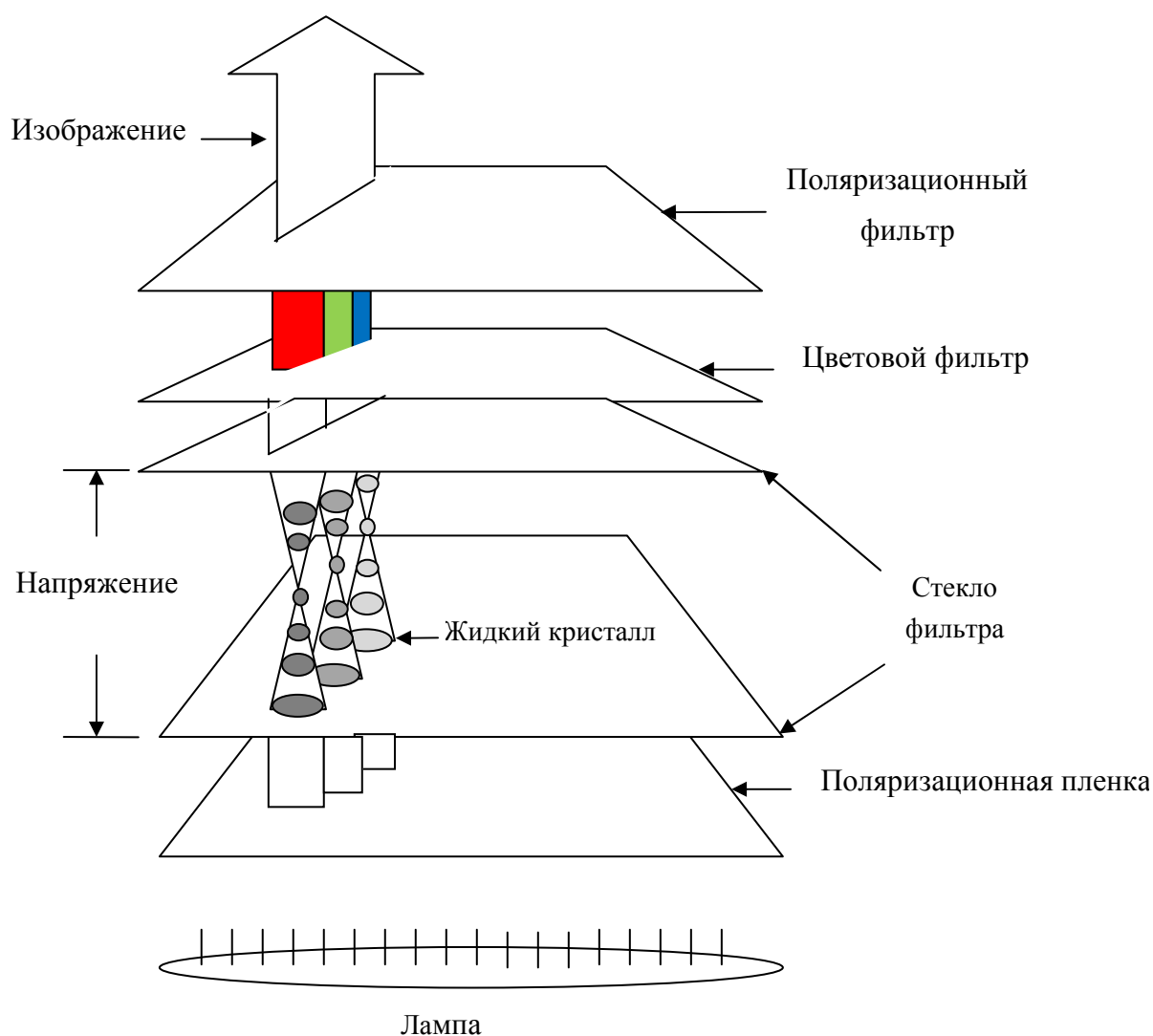


Рис. 60. Устройство ЖК-монитора

Жидкие кристаллы образуют матрицу жидкокристаллического экрана. Типы LCD определяются способом организации кристаллов данной матрицы и управления ими при подаче напряжения.

Как и в CRT, пиксель формируется из трех участков – красного, зеленого и синего. А различные цвета получаются в результате изменения величины соответствующего электрического заряда.

Так как LCD технология адресует каждый пиксель отдельно, четкость получаемого изображения выше в сравнении с CRT.

Основные характеристики мониторов

Размер экрана, имеющего прямоугольную форму, обычно измеряется по диагонали в дюймах (12”, 14”, 15”, 17”, 19”, 21”, ...).

Разрешающая способность. Количество пикселей по вертикали (в строке) и горизонтали (в столбце). Разрешение экрана определяет четкость текста и изображений, отображаемых на экране. При больших разрешениях, например при 1280 x 1024 пикселей, объекты выглядят четче. Кроме того, объекты оказываются мельче, и на экране их может поместиться больше. При низких разрешениях, например при 800 x 600 пикселей, на экране умещается меньше объектов, но они выглядят более крупными. При выборе разрешения, которое не поддерживается монитором, экран на несколько секунд становится черным, пока восстанавливается исходное разрешение монитора.

Соотношение сторон экрана — стандартный (4:3), широкоформатный (16:9, 16:10) или другое соотношение (например 5:4).

Размер пикселя (или зерна).

В нижеследующей таблице приведены примеры соотношения приведенных выше характеристик мониторов.

Диагональ матрицы, дюймы	Разрешение в пикселях	Соотношение сторон	Размер пикселя, мм
15	1024 x 768	4:3	0,297
16	1280 x 1024	5:4	0,248
17	1280 x 768	15:9	0,2895
17	1280 x 1024	5:4	0,264
17	1440 x 900	16:10	0,255
18,1	1280 x 1024	5:4	0,2805
19	1280 x 1024	5:4	0,294
19	1440 x 900	16:10	0,284
19	1600 x 1200	4:3	0,242
20,1	1680 x 1050	16:10	0,258
20,1	1600 x 1200	4:3	0,255
20,8	2048 x 1536	4:3	0,207
21,3	1600 x 1200	4:3	0,27
22	1680 x 1050	16:10	0,282
22,2	3840 x 2400	16:10	0,1245
23	1920 x 1200	16:10	0,258
23,1	1600 x 1200	4:3	0,294
24	1920 x 1200	16:10	0,269
26	1920 x 1200	16:10	0,2865
27	1920 x 1200	16:10	0,303

30	2560 x 1600	16:10	0,251
----	-------------	-------	-------

Глубина цвета — количество бит на кодирование одного пикселя (от монохромного до 32-битного).

Частота обновления экрана (Гц).

Для LCD-мониторов эта характеристика не является критичной, так как изображение, если меняется, то меняется целиком. Нормальная частота обновления экрана – 60Гц.

Изображение в CRT-мониторах меняется последовательно по одному пикселю (горизонтальная и вертикальная развертки), поэтому частота обновления должна быть не менее 85Гц, в противном случае изображение на экране постоянно мерцает, что чрезвычайно вредно для глаз.

Время отклика пикселей. Минимальное время, необходимое пикселю для изменения своей яркости. Важная характеристика LCD-мониторов. Состояние пикселя в ЖК-матрице меняется за счет изменения угла поворота жидких кристаллов под действием приложенного к ним электрического поля. Однако жидкие кристаллы - вещество сравнительно вязкое, поэтому поворот происходит не мгновенно, а за достаточно большое время порядка единиц миллисекунд. Чтобы при быстрой смене картинки не было «смазанностей», время отклика должно быть менее 8 миллисекунд,

большее время отклика может повлиять на степень «размытости» изображения.

Угол обзора. Угол, при котором падение контраста достигает заданного. Если изображение на ЭЛТ практически не страдает даже при взгляде почти параллельно плоскости экрана, то для первых поколений небольшие углы обзора были серьезной проблемой: даже небольшое отклонение от перпендикуляра приводило к заметному падению контрастности и искажению цветопередачи. Со временем эта проблема была снята и приемлемый угол обзора приближается к 180°. Наиболее удобно пользоваться мониторами у которых угол обзора составляет по вертикали и по горизонтали 160 градусов.

Контрастность. Данная характеристика показывает отношение яркостей белого цвета к черному цвету. Указывается в технических характеристиках как пропорция 800:1, 500:1 и т.д.

Яркость. Для удобства работы и комфорта яркость монитора должна быть от 80 кандел на квадратный метр (указывается как 80 кд/кВ.м).

Интерфейсы подключения

1. **Разъем VGA** (Video Graphics Array) – аналоговый стандарт, предназначенный для мониторов с расширением 640*480. При увеличении разрешения, качество цифровой картинки ухудшается. Для получения изображения высокого качества требуются разъемы цифрового стандарта.

2. **DVI** (Digital Visual Interface). Цифровой интерфейс передает видеосигнал в цифровом формате и обеспечивает высокое качество цифрового изображения. Имеет

совместимость с аналоговым разъемом VGA (передает одновременно сигнал и в цифровом формате, и в аналоговом).

Одноканальная модификация обеспечивает разрешение монитора 1920*1080.

Более дорогие модели снабжены двухканальным интерфейсом и могут поддерживать разрешение до 2560*1600. Для ноутбука разработан интерфейс mini-DVI.

3. **HDMI** (High Definition Multimedia Interface) - мультимедийный цифровой интерфейс с высоким разрешением.

4. **DP** (DisplayPort). Интерфейс позволяет передавать данные цифровыми пакетами без утери качества. Данный разъем призван заменить стандарт DVI. Особенностью интерфейса является то, что он позволяет подключать несколько мониторов, соединённых последовательно, к одному системному блоку.

По данным тестирования VGA-интерфейсы показывают самое низкое качество отображения. Для монитора с диагональю более 17" и разрешением более 1024*786, лучше использовать разъемы DVI, HDMI, DP.

Видеосистема компьютера

Видеосистема компьютера состоит из трех компонент:

- **монитор;**
- **видеокарты;**
- **программное обеспечение** (драйверы видеосистемы).

Видеокарта. Назначение и состав

Связь ЭВМ с монитором осуществляется с помощью **видеокарты** (графической карты, видеоадаптера, графического адаптера, графической платы).

Ранее видеокартой называли устройство, преобразующее графический образ, хранящийся, как содержимое памяти компьютера или самого адаптера, в иную форму, предназначенную для дальнейшего вывода на экран монитора. В настоящее время эта функция утратила основное значение, и в первую очередь под графическим адаптером понимают устройство, занимающееся формированием самого графического образа.

Современные видеокарты не ограничиваются простым выводом изображения, они имеют встроенный графический процессор, который может производить дополнительную обработку, снимая эту задачу с центрального процессора компьютера.

Замечание. В последнее время также имеет место тенденция использовать вычислительные возможности графического процессора для решения неграфических задач

Современная видеокарта состоит из следующих частей:

- **графический процессор (GPU)** — занимается расчётами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчёты для обработки команд трёхмерной графики. Является основой графической платы, именно от него зависят быстродействие и возможности всего устройства. Современные графические процессоры по сложности мало чем уступают центральному процессору компьютера, и зачастую превосходят его как по числу транзисторов, так и по вычислительной мощности, благодаря большому числу универсальных вычислительных блоков.

- **видеоконтроллер** — отвечает за формирование изображения в видеопамяти, даёт команды на формирование сигналов развёртки для монитора и осуществляет обработку запросов центрального процессора. Кроме этого, обычно присутствуют контроллер внешней шины данных, контроллер внутренней шины данных и контроллер видеопамяти.

- **видеопамять** — выполняет роль кадрового буфера, в котором хранится изображение, генерируемое и постоянно изменяемое графическим процессором и выводимое на экран монитора. В видеопамяти хранятся также промежуточные невидимые

на экране элементы изображения и другие данные. Видеопамять бывает нескольких типов, различающихся по скорости доступа и рабочей частоте.

Помимо видеопамяти, находящейся на видеокарте, современные графические процессоры обычно используют в своей работе часть общей системной памяти компьютера, прямой доступ к которой организуется драйвером видеоадаптера.

Замечание. В некоторых архитектурах в качестве видеопамяти используется часть системной памяти компьютера.

- **цифро-аналоговый преобразователь (RAMDAC — Random Access Memory Digital-to-Analog Converter)** — служит для преобразования изображения, формируемого видеоконтроллером, в уровни интенсивности цвета, подаваемые на аналоговый монитор. Возможный диапазон цветности изображения определяется только параметрами. Чаще всего RAMDAC имеет четыре основных блока: три цифроаналоговых преобразователя, по одному на каждый цветовой канал (красный, зелёный, синий - RGB), и SRAM (статическую память) для хранения данных. Большинство RAMDAC имеют разрядность 8 бит на канал — получается по 256 уровней яркости на каждый основной цвет, что в сумме дает 16,7 млн цветов. Некоторые RAMDAC имеют разрядность по каждому каналу 10 бит (1024 уровня яркости), что позволяет сразу отображать более 1 млрд цветов, но эта возможность практически не используется.

- **видео-ПЗУ (Video ROM)** — постоянное запоминающее устройство, в которое записаны видео-BIOS, экранные шрифты, служебные таблицы и т. п. ПЗУ не используется видеоконтроллером напрямую — к нему обращается только центральный процессор. Хранящийся в ПЗУ видео-BIOS обеспечивает инициализацию и работу видеокарты до загрузки основной операционной системы, а также содержит системные данные, которые могут читаться и интерпретироваться видеодрайвером в процессе работы (в зависимости от применяемого метода разделения ответственности между драйвером и BIOS).

- система охлаждения — предназначена для сохранения температурного режима видеопроцессора и видеопамяти в допустимых пределах.

Уровни программного обеспечения машинной графики

Программное обеспечение машинной графики состоит из трех уровней.

Первый уровень составляют программы видеопроцессора, которые в качестве исходных данных имеют информацию, записанную в экранной памяти, и организуют ее сканирование для управления процессом регенерации. Эти программы обычно жестко запаяны и реализованы в виде микропрограмм. Их разработка относится к компетенции проектировщиков видеосистем. При этом очень важна структура информации в экранной памяти.

Второй уровень связан с процедурами формирования кадра экранной памяти. Исходными данными для этих программ являются команды примитивов машинной графики. Это команды типа: нарисовать отрезок, начертить окружность, изменить цветовую гамму фона, поместить литеру и т.д.

Программы формирования нового кадра, анализируя параметры соответствующих команд, вносят изменения в экранную память или формируют ее заново. Поскольку кодировка экранной памяти в очень большой степени зависит от аппаратуры видеопроцессора, этот уровень программ является машинно-зависимым. Существует несколько стандартов задания графических примитивов в памяти ЭВМ. Такие стандарты представляют собой своего рода машинно-независимый язык задания графических объектов, возможно, встроенный в универсальный язык программирования в виде библиотеки специальных операторов графики.

Третий уровень связан с процедурами формирования динамической последовательности смены кадров или сюжета процесса визуализации. Он имеет дело с преобразованием графических объектов, их объединением, вращением, уменьшением,

компоновкой во времени и пространстве. Исходными данными для этих программ являются воображение и творчество пользователя.

Другие виды мониторов

Плазменные дисплеи. В основе – возможность управлять возникновением электрических разрядов в некоторых газах и сопровождающим их свечением. Такие дисплеи обладают высоким качеством изображения и могут иметь значительно большие, чем у привычных компьютеров, размеры экранов при небольшой толщине (экран с диагональю около 1 м при толщине 8-10 см).

Достоинствами плазменной панели являются следующие свойства:

- как и в ЖК-панелях, в плазменных экранах отсутствует мерцание изображения,
- картинка имеет одинаковую высокую четкость по всему рабочему полю,
- малая толщина панели (не более 6 дюймов),
- бытовые дисплеи можно вешать на стенку,
- прекрасная обзорность (под любым углом),
- высокая контрастность.

К основным недостаткам такого типа мониторов можно отнести:

- достаточно высокую потребляемую мощность, возрастающую при увеличении диагонали экрана,
- низкую разрешающую способность (не более 1024 x 768), обусловленную большим размером элемента изображения.

Сенсорный экран — координатное устройство, позволяющее путем прикосновения к области экрана монитора производить выбор необходимого элемента данных, меню или осуществлять ввод данных в ЭВМ. Сенсорные экраны наиболее пригодны для организации гибкого интерфейса, интуитивно понятного даже далеким от техники пользователям. С распространением карманных, планшетных компьютеров, устройств для чтения электронных книг и различных терминалов, сенсорные экраны стали привычными и незаменимыми и область их использования постоянно расширяется.

Рассмотрение более подробной информации о плазменных, сенсорных и других типах дисплеев в настоящем пособии не предусматривается.

Синтезаторы звука – своеобразные устройства вывода. Простейшие из них есть в арсенале почти у всех персональных компьютеров и представляют собой обычный малогабаритный динамик, напряжение сигнала на котором с большой частотой изменяется компьютером. Таким способом удается подать простой звуковой сигнал, указывающий на наступление какого-либо события. Многие языки программирования дополняются командами типа BEEP, SOUND, позволяющими программировать серии звуков. Если звукогенератор физически реализован так, что частота звучания поддается регулированию, то можно запрограммировать несложную мелодию, а если есть несколько независимых звукогенераторов, то – и звучание оркестра. Для этого в современных компьютерах устанавливается специальная плата – звуковая карта, – способная преобразовывать аналоговый звуковой сигнал в последовательность двоичных цифр и наоборот. Существуют и синтезаторы речи, назначение которых понятно из названия.

Основным параметром звуковой карты является разрядность, определяющая количество битов, используемых при преобразовании сигналов из аналоговой в цифровую форму и наоборот. Чем выше разрядность, тем меньше погрешность, связанная с оцифровкой, тем выше качество звучания. Минимальным требованием сегодняшнего дня являются 16 разрядов, а наибольшее распространение имеют 32-разрядные и 64-разрядные устройства.

В области воспроизведения звука наиболее сложно обстоит дело со стандартизацией. Отсутствие единых централизованных стандартов привело к тому, что

ряд фирм, занимающихся выпуском звукового оборудования, де-факто ввели в широкое использование свои внутрифирменные стандарты.

10.4.3. Устройства обмена данными (средства телекоммуникации)

Модем. Устройство, предназначенное для обмена информацией между удаленными компьютерами по каналам связи (МОдулятор + ДЕМОдулятор) с использованием телефонной сети.

Модем работает следующим образом: принимая от ПК данные, он преобразует их в аналоговый сигнал и передает в канал связи. В модеме принимающего ПК происходит обратное преобразование - сигнал преобразуется в цифровой код.

К основным потребительским параметрам модемов относятся:

- производительность (бит/с);
- поддерживаемые протоколы связи и коррекции ошибок;
- шинный интерфейс, если модем внутренний.

От производительности модема зависит объем данных, передаваемых в единицу времени.

Сетевой адаптер - это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы и распределение функций между сетевым адаптером и драйвером может изменяться от реализации к реализации

Сетевой адаптер обычно выполняет следующие функции:

- Оформление передаваемой информации в виде **кадра** определенного формата.
- Получение доступа к среде передачи данных.
- Кодирование последовательности бит кадра последовательностью электрических сигналов при передаче данных и декодирование при их приеме.
- Преобразование информации из параллельной формы в последовательную и обратно. Эта операция связана с тем, что для упрощения проблемы синхронизации сигналов и удешевления линий связи в вычислительных сетях информация передается в последовательной форме, бит за битом, а не побайтно, как внутри компьютера.
- Синхронизация битов, байтов и кадров. Для устойчивого приема передаваемой информации необходимо поддержание постоянного синхронизма приемника и передатчика информации. Сетевой адаптер использует для решения этой задачи специальные методы кодирования.

Интерактивные устройства (интерактивные доски, интерактивные проекторы и интерактивные приставки).

Все три типа приборов решают основную задачу – постоянное взаимодействие докладчика с компьютером во время проведения презентации и показа видеоматериала. В этом – их основное назначение и сходство.

11. Устройство персонального компьютера

Рассмотрим устройство компьютера на примере самой распространенной компьютерной системы — персонального компьютера.

Персональным компьютером (ПК) называют сравнительно недорогой универсальный микрокомпьютер, рассчитанный на одного пользователя. Персональные компьютеры обычно проектируются на основе принципа *открытой архитектуры*.

Принцип открытой архитектуры заключается в регламентации и стандартизации только описания принципа действия компьютера и его конфигурации (определенная совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.

Компьютер легко расширяется и модернизируется за счёт наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

11.1. Состав компьютера

Современный персональный компьютер состоит из нескольких основных конструктивных компонент:

- системного блока;
- монитора;
- клавиатуры;
- манипуляторов.

В системном блоке размещаются:

- блок питания;
- накопитель на жёстких магнитных дисках;
- системная плата;
- платы расширения;
- накопитель CD-ROM;
- и др.

Основные электронные компоненты, определяющие архитектуру процессора, размещаются на основной плате компьютера, которая называется **системной** или **материнской**:

- центральный процессор;
- постоянная и оперативная память, кэш-память;
- интерфейсные схемы шин;
- гнезда расширения;
- обязательные системные средства ввода-вывода
- и др.

Системные платы исполняются на основе наборов микросхем, которые называются **чипсетами**. Часто на системных платах устанавливают и контроллеры дисковых накопителей, видеоадаптер, контроллеры портов и др.

11.2. Контроллеры и адаптеры. Общий состав устройств

Основной состав устройств ПК представлен в нижеследующей таблице, а упрощенная схема их взаимосвязей - на рис.61.

Компоненты персонального компьютера	
Системный блок	Блок питания • Система охлаждения • Материнская плата •Центральный процессор • Шины • Видеокарта • Звуковая плата • Сетевая плата
Запоминающие устройства	
Без механики	Оперативная память • Твердотельный накопитель (Флеш-память)

С механикой	Жесткий диск • Стример • Оптический привод (CD • DVD)
Устройства вывода	Звук: (Акустика • Динамик) Изображение: (Монитор • Графопостроитель (плоттер) • Принтер)
Устройства ввода	Клавиатура • Мышь • Трекбол • Тачпад • Сенсорный экран Звук: (Микрофон) Изображение: (Световое перо • Графический планшет • Сканер • Веб-камера) Игровые устройства (Джойстик)
Прочее	Модем

Контроллеры и адаптеры дополнительных устройств, либо сами эти устройства, выполняются в виде **плат расширения**, которыми снабжаются устройства компьютера с целью совместимости их интерфейсов, и подключаются к шине с помощью **разъемов расширения**, называемых также **слотами расширения** (например, модем, сетевая плата, видеоплата и т.д).

Контроллеры осуществляют и непосредственное управление периферийными устройствами по запросам микропроцессора.

Порты устройств представляют собой некие электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора.

Портами также называют устройства стандартного интерфейса - **последовательный, параллельный и игровой** порты (или интерфейсы):

- **Последовательный** порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. К последовательному порту обычно подсоединяют медленно действующие или достаточно удалённые устройства, такие, как мышь и модем.
- **Параллельный** порт получает и посылает данные побайтно. К параллельному порту подсоединяют более "быстрые" устройства — принтер и сканер.
- Через **игровой** порт подсоединяется джойстик.

Клавиатура и монитор подключаются к своим **специализированным** портам, которые представляют собой просто **разъёмы**.

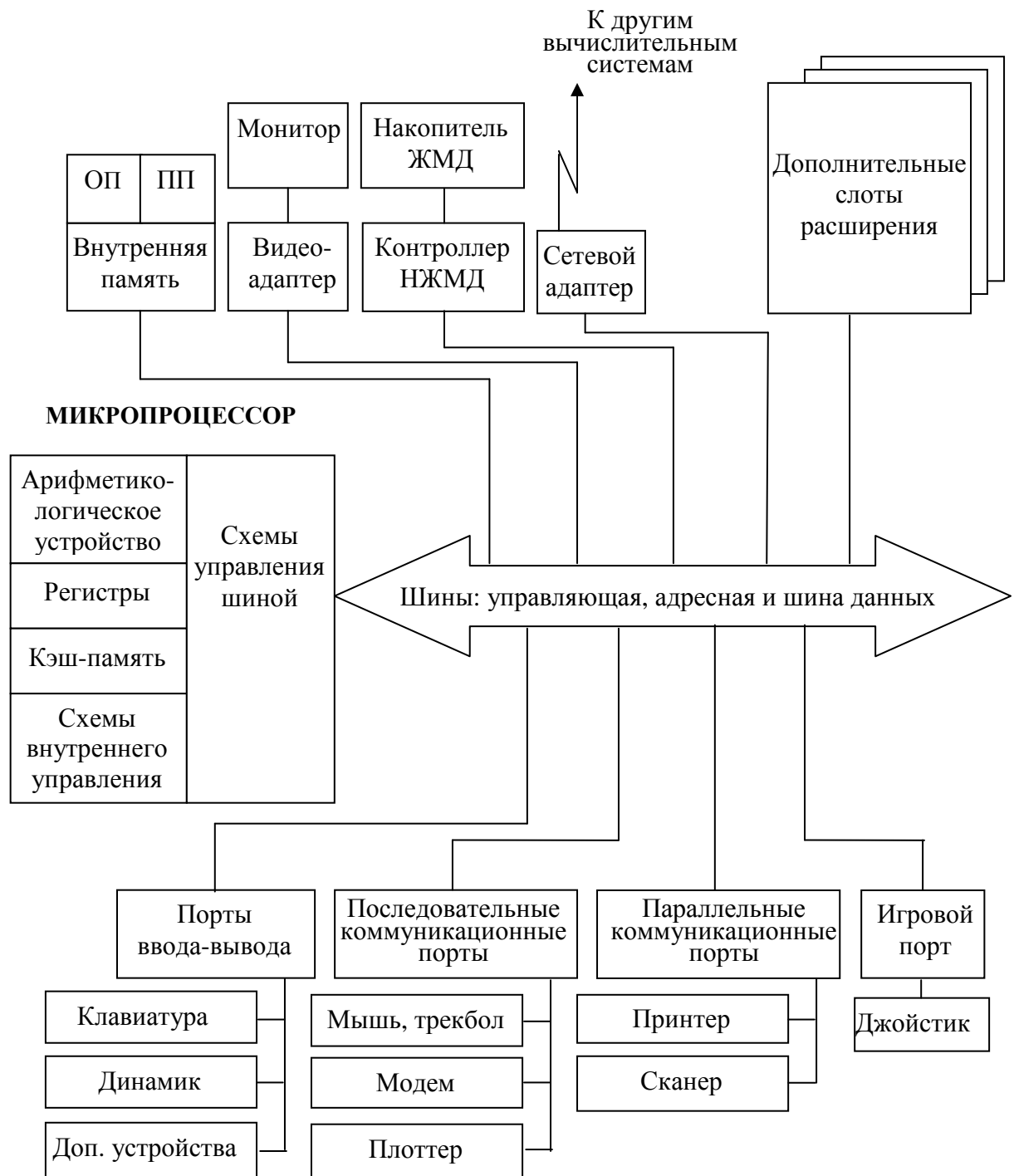


Рис.61. Общая структура персонального компьютера с подсоединенными периферийными устройствами