

Бесплатно

87-4

22389
-1

Министерство высшего и среднего специального образования
Латвийской ССР

Латвийский ордена Трудового Красного Знамени
государственный университет им.П.Стучки

Вычислительный центр

Р. В. ФРЕЙВАЛД, А. А. КАЛИС, С. И. ПАВЛОВ,
Д. Я. ТАЙМИНЯ

ПОНЯТИЕ О РАБОТЕ Э В М

Методические указания

Латвийский государственный университет им.П.Стучки
Рига 1986

Издание представляет собой первую часть методических указаний под редакцией Р.В.Фрейвалда для учителей средних школ, готовящих уроки по новому учебному предмету "Основы информатики и вычислительной техники" с использованием второй части пробного учебного пособия под редакцией А.П.Ершова и В.М.Монахова. Указания могут быть полезны и студентам специальности 2013 - математика во время педагогической практики и при изучении спецкурса. В издании излагаются основные методические принципы построения нового учебного курса, даются комментарии к наиболее сложным местам учебного пособия, приводятся решения упражнений и дополнительные упражнения с решениями.

Издание подготовлено коллективом авторов:

Р.В.Фрейвалд - доктор физ.-мат.наук, зам.директора ВЦ,

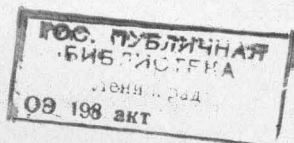
А.А.Калис - мл.научный сотрудник,

С.И.Павлов - кандидат физ.-мат.наук, ст.научный сотрудник,

Д.Я.Тайминя - аспирантка.

УТВЕРЖДЕНЫ

на заседании секции по вопросам математического обеспечения
Совета Вычислительного центра ЛГУ им.П.Стучки
18 июня 1986 г., протокол № 3.



ВВЕДЕНИЕ

В издательстве "Просвещение" издана вторая часть пробного учебного пособия по новому школьному предмету "Основы информатики и вычислительной техники" под редакцией А.П.Ершова и В.М.Монахова. Это пробное учебное пособие (далее в тексте сокращенно именуемое учебником) предназначено для учеников 10 класса общеобразовательных школ для продолжения изучения одноименного учебного предмета в безмашином варианте. Ученики, изучающие этот предмет в машинном варианте, изучат материал этого учебника и еще дополнительный материал о той ЭВМ, к которой они будут иметь доступ.

Наш учебный предмет вводился в школы быстрыми темпами. Когда ученики приступили к изучению материала 9 класса, учебник 10 класса еще не был написан. Не было и опыта преподавания таких вопросов.

При написании учебника 10 класса уже можно было использовать некоторый накопившийся опыт. Неудивительно, что взгляды авторов учебника на некоторые вопросы изложения материала претерпели изменения. По сравнению с наметенным ранее, был уменьшен материал о физических основах устройства ЭВМ. Меньше место отведено изложению программирования в машинных кодах. Соответственно в учебник включен дополнительный материал о составлении алгоритмов на алгоритмическом языке.

В мире имеется много тысяч различных языков программирования. Около десятка из них вполне доступны для изучения в школе. Хотя учебник предназначен для безмашиного варианта обучения, все же нужно было выбрать, с каким языком программирования ученики будут знакомиться более подробно. Конечно, доступность языка не единственный критерий отбора. Нужно было учитывать также перспективность этого языка, стараясь прогнозировать, будут ли развиваться идеи, заложенные в язык. Авторы учебника выбрали



Рапиру и Бейсик. Эти языки довольно просты и распространены. Хотя бы один из них реализован на почти каждом типе ЭВМ, который может быть доступен школе.

Предполагается, что ученики будут учить один из языков Рапира или Бейсик (по усмотрению учителя), а не оба.

К сожалению, в некоторое нарушение этого принципа, авторы отвели Рапире и Бейсику неравноценный объем текста. Поэтому мы сильно увеличили раздел методических указаний, посвященный Бейсику. Теперь учитель в случае необходимости сможет подчеркнуть здесь все нужные сведения о Бейсике, не обращаясь к более специальным книгам.

Следует также отметить, что вариант языка программирования Рапира, принятый в учебнике, является модернизацией Рапиры, описанной в журнале "Квант" № 4-3, 1980. Этот вариант отличается также от варианта Рапиры, реализованной в школьной ЭВМ "АГАТ".

Для помощи учителям, готовящим уроки по вышеупомянутому учебнику подготовлено настоящее методическое пособие. Структура этого пособия согласована со структурой учебника. По каждому параграфу дается основной текст, в котором поясняются принципы построения учебного курса, обращается внимание на трудные для понимания аспекты учебного материала, предсказываются некоторые возможные ошибки учеников, а также приводятся ответы на вопросы и решения упражнений из учебника и в некоторых случаях приводятся дополнительные упражнения и решения к ним.

Как в учебнике, так и в настоящем методическом пособии особняком стоит раздел "Задачи на повторение". Здесь собраны задачи повышенной трудности по всему учебному материалу. В нашем методическом пособии даны не только решения этих упражнений на алгоритмическом языке, но и проверенные на машине решения этих упражнений на языках программирования Рапира и Бейсик. Учитель, имеющий доступ к ЭВМ, сможет использовать эти программы для доказательства, как ЭВМ решает конкретные задачи. Так как в тексте учебника подробно объясняются только некоторые коман-

ды ЭВМ ДВК-2М, то в приложении методического пособия приводится и полный список этих команд. Такая информация полезна для понимания уровня сложности команд ЭВМ, для понимания, каких команд ЭВМ заведомо не имеет. Приводятся также дополнительные сведения об устройстве ЭВМ, об использовании программируемых калькуляторов, о версиях языка программирования Бейсик, которые реализованы на доступных школе ЭВМ.

Авторы выражают глубокую благодарность многочисленным коллегам, которые дали полезные советы в процессе работы над методическим пособием и, особенно, Анатолию Георгиевичу Кушниренко (МГУ им. М.В. Ломоносова), конструктивная критика которого привела к значительной переработке первоначального текста.

§1. ОБЩАЯ СХЕМА УСТРОЙСТВА ЭВМ (Ич.)

Материал этого параграфа и предшествующего ему введения в раздел I носит скорее ознакомительный, чем учебный характер. Предполагается, что этот урок будет иметь характер лекции. С одной стороны, здесь повторяются некоторые понятия, которые упоминались во введении учебника 9-го класса (бит, байт, процессор и др.). Правда, нельзя рассчитывать, что ученики это помнят. Ведь далее в течении курса 9 класса эти понятия не повторялись. С другой стороны, происходит углубление пройденного материала: появляются новые понятия (адрес, машинное слово), даются первые сведения о взаимодействии отдельных частей машины.

Основная цель. Дать ученику самое общее представление о том, из каких частей состоит ЭВМ, и, что эти части делают, и как они взаимодействуют.

Требования к знаниям и умениям. Ученики должны знать основные функции процессора, памяти, внешних устройств ЭВМ и понимать, как они взаимодействуют друг с другом через магистраль.

М е т о д и ч е с к и е у к а з а н и я

Если учитель имеет возможность организовать экскурсию в вычислительный центр, то желательно рассказать этот материал непосредственно перед экскурсией. Тогда ученики лучше поймут ЭВМ, поймут, что и большие и малые ЭВМ состоят из таких же составных частей: из процессора, памяти, внешних устройств.

Разумеется, ЭВМ, которая установлена в вычислительном центре, будет выглядеть иначе, чем ДВК-2М, изображенная на рис. учебника. Большие ЭВМ, мини-ЭВМ и микро-ЭВМ отличаются не просто размерами машины. С другой стороны, невозможно изучать устройство ЭВМ вообще, а при том малом учебном времени, что отведено на изучение этой темы, невозможно даже сравнить различные классы ЭВМ. Остается один путь: выбрать одну конкретную ЭВМ и весь ма-

териал раздела I рассмотреть применительно к этой ЭВМ, что и сделано в учебнике.

ЭВМ ДВК-2М нельзя назвать ни самой распространенной, ни в каком-либо другом смысле самой популярной советской ЭВМ. Почему же в учебнике выбрана была именно эта машина?

Во-первых, нужно было рассматривать именно микро-ЭВМ, так как они по устройству проще, и, кроме того, во всем мире в школу приходят именно микро-ЭВМ. Было бы странно, устанавливать в школах ЭВМ одного класса, а в учебнике описывать устройство ЭВМ другого класса.

Во-вторых, среди советских микро-ЭВМ нет ни одной, которая была бы по всем параметрам неоспоримо лучше, чем ДВК-2М. Более того, ЭВМ ДВК-2М можно считать "типичным представителем" своего класса, так как микропроцессор К1801/ВМ1, используемый в этой ЭВМ, используется также и в отечественном Бытовом Компьютере ЭЛЕКТРОНИКА БК-0010.

Модернизация К1801/ВМ1, а именно микропроцессора К1801/ВМ1, будет использована в массовом школьном компьютере УК-ИЧ. (В §§2-4 описывается работа процессора. Отметим, что упомянутый модернизированный процессор имеет все возможности старого процессора, но, кроме того, еще и некоторые дополнительные.) Наконец, в микропроцессоре К1801/ВМ1 используется ряд прогрессивных идей, используемых и во многих других современных микропроцессорах.

Конечно, не предполагается, что ученики будут заучивать наизусть названия микропроцессоров, будущих школьных компьютеров и т.д.

Много дополнительного материала для подготовки к уроку учитель может найти в приложении III к учебнику. В частности, в п.5 этого приложения объясняется важный магистрально-модульный принцип построения ЭВМ, о котором желательно упомянуть на уроке.

К о м м е н т а р и и к р и с у н к а м у ч е б н и к а

1) Оба вертикальных ряда элементов (микросхем) на левой части рис.4 - это память ЭВМ.

2) Изображенная на рис.5 ЭВМ ДВК-2М - хорошая машина,

но по размерам немного великовата, в частности, занимает много места на столе. Школьные ЭВМ будут меньше, хотя и ничуть не менее мощные. За счет чего это станет возможным? Дело в том, что ЭВМ ДВК-2М была разработана и начала выпускаться несколько лет назад. За прошедшие годы техника шла вперед и это дало возможность сделать электронные устройства более компактными: те функции, которые раньше выполняла целая группа микросхем, теперь может выполнить одна микросхема. Технический прогресс приводит к улучшению характеристик изделий во всех отраслях народного хозяйства: сравните, например, вес цветных телевизоров, выпускаемых сейчас и десять лет назад. Однако в области вычислительной техники этот прогресс сейчас наиболее заметен.

3) Микропроцессор на рис.6 изображен с небольшим увеличением. На самом деле он примерно в 1,5 раза меньше.

Д о м а ш н е е з а д а н и е : §1, №1 §1.

ОТВЕТЫ НА ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Что такое ЭВМ? Для чего она используется?

Ответ. ЭВМ - это машина для хранения и автоматической обработки информации. ЭВМ используется в обработке результатов экспериментов, в обучении, в автоматическом проектировании, в справочно-информационных системах.

2. Перечислите внешние устройства ЭВМ. Расскажите об их назначении.

Ответ. Внешними устройствами ЭВМ являются клавиатура, дисплей, алфавитно-цифровое печатающее устройство, дисковод. Могут быть еще другие внешние устройства, например, магнитофон, графопостроитель. Внешние устройства ЭВМ обеспечивают ввод (клавиатура) и вывод (дисплей, АЦПУ, графопостроитель) информации, взаимодействие ЭВМ с человеком и другими ЭВМ. Дисковод, магнитофон позволяет организовать хранение информации на внешних носителях - гибких дисках и кассетах магнитофона.

ВОПРОСЫ И ОТВЕТЫ

1. Что такое микросхема, микропроцессор?

Ответ. Информация, циркулирующая внутри ЭВМ, кодируется с помощью электрических сигналов. Электронные устройства, осуществляющие обработку этих сигналов, и состоящие из тысяч и даже сотен тысяч согласованно работающих элементов, и размещенные в небольшом объеме, называются микросхемами. Если микросхема содержит целиком весь процессор ЭВМ, то она называется микропроцессором.

2. Что такое магистраль? Для чего она служит?

Ответ. Магистраль можно представить как пучок проводов, к которому параллельно подсоединены все компоненты ЭВМ. Магистраль служит для осуществления связи и обмена информацией между компонентами ЭВМ.

3. Как происходит чтение слова из памяти?

Ответ. Пусть процессору необходимо прочесть слово из памяти. Чтение происходит следующим образом. 1) Процессор передает на магистраль адрес нужного слова и указание "читай слово". 2) Память считывает адрес с магистрали и передает на магистраль содержимое соответствующего слова. 3) Процессор считывает слово с магистрали.

УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Память ЭВМ ДВК-2М позволяет хранить 56 килобайтов. Учитывая, что 1 килобайт = 1024 байта, определите, сколько слов может поместиться в памяти этой ЭВМ.

Решение. Память ЭВМ содержит $56 \cdot 1024 = 57344$ байта. Машинное слово содержит два байта. Следовательно, в ЭВМ может поместиться $57344 : 2 = 28672$ слов.

§ 2. ОСНОВНОЙ АЛГОРИТМ РАБОТЫ ПРОЦЕССОРА (I ч.)

Основная цель. Ознакомить учеников работой процессора ЭВМ ДВК-2М, его командами, степенью сложности этих команд и данных, которые обрабатываются этими командами.

Требования к знаниям и умениям. Ученики должны знать понятия регистра и счетчика команд, знать порядок выполнения процессором команд. Ученики должны уметь изобразить выполнение простейших программ в машинных командах при помощи таблицы значений.

Методические указания

Устройство процессора ЭВМ нас интересует прежде всего с точки зрения того, как процессор работает, как он производит обработку информации. (Физические основы устройства ЭВМ будут рассмотрены далее в § 5, да и то очень коротко).

Процессор ЭВМ ДВК-2М имеет небольшую собственную память, состоящую из 8 машинных слов (по 16 битов каждое), называемых регистрами, и еще 2 выделенных бита, называемых N и Z . Общая память ЭВМ значительно больше. Как указано в упражнении I из § 1, память ЭВМ ДВК-2М позволяет хранить 56 кбайтов.

Надо обратить внимание на то, что адреса слов всегда четные числа, а каждая команда занимает одно слово. Поэтому и адреса команд - всегда четные числа, например, 1500, 1502, 1504.

Счетчик Команд (т.е. регистр $R7$) всегда содержит номер той команды, которую процессор должен выполнить следующей.

Действия процессора по выполнению очередной команды состоят из четырех этапов. Процессор

- 1) читает адрес из Счетчика Команд;
- 2) читает слово из памяти ЭВМ по этому адресу;

3) увеличивает Счетчик Команд на 2;

4) выполняет команду, записанную в прочитанном слове.

Все же работу процессора нельзя объяснить, изучая только отдельные команды и не разбирая работу целых программ. Поэтому в §§ 2-4 много простейших программ разбирается очень подробно. Важно отметить, что это делается исключительно для объяснения работы ЭВМ. Совсем не предполагается довести навыки учеников по программированию в машинных командах до какого-то технического уровня.

Программировать - это означает записывать алгоритм в виде четко определенной последовательности операций (команд), которые исполнитель может выполнить механически, не заботясь о смысле этих операций (команд). При понимании работы ЭВМ важно различать три различных уровня сложности таких элементарных операций:

1) С самими простыми элементарными операциями имеют дело инженеры, конструирующие микропроцессор. Здесь приходится иметь дело с элементарными порциями информации - битами - и действиями над ними. Несколько таких операций и принципы их физической реализации описаны в § 5. Конструктор процессора должен добиться, чтобы при всевозможных вариантах подачи на входы процессора нулей и единиц на выходы появлялись правильные результаты. При выполнении одной машинной команды выполняется очень много таких операций, но того, кто пользуется микропроцессором, эти операции обычно уже не интересуют.

2) В § 17 учебника будет рассказано о том, что для работы ЭВМ необходимо специальное программное обеспечение, в частности, так называемая операционная система. Для разработчика операционной системы элементарной командой является как раз команда процессора, а заметная по объему часть операционной системы пишется в машинных командах или на близком к машинным командам специальном языке Ассемблер.

3) Как только операционная система создана, элементарными можно считать команды операционной системы, каждая из которых может выражаться сотнями и тысячами машинных команд. Таким же образом, после создания транслятора с языка программирования элементарной командой можно считать команду на языке программирования или вызов вспомогательного алгоритма. Для реализации одной такой команды может потребоваться довольно сложная программа в машинных командах.

В течение определенного времени (в месяц, год или неделю) программист может составить программу, содержащую только ограниченное число команд. Удивительный феномен программирования состоит в том, что это число команд зависит от физической работоспособности программиста, ясности мышления и т.д., но не зависит от уровня детализации элементарных операций, которыми пользуется программист (микрокоманды, машинные команды, команды на языке Ассемблера или команды на языках программирования высокого уровня). Поэтому, планируя разработку программы для ЭВМ, всегда нужно четко представлять, что нам важнее: подготовить программу быстро и с меньшим числом программистов (но программу, работающую немного медленнее и занимающую больше памяти ЭВМ), или потрудиться над программой дольше с участием большего числа программистов и сделать очень экономную программу.

Пользователи ЭВМ обычно выбирают первый путь и экономят труд людей за счет большего использования машин. Поэтому и в нашем учебном предмете основной упор сделан на алгоритмический язык и на языки программирования Рэпир и Бейсик (они, правда, представляют только очень малую часть тех удобств для программиста, которые характеризуют языки программирования высокого уровня). Все же надо, чтобы ученики хотя бы в самой малой мере почувствовали возможность другого пути. Правда, этот другой путь используется очень редко, и программирование в машинных командах или на Ассемблере обычно остается уделом лишь

разработчиков так называемых системных программ.

Принцип изложения материала в §§ 2-4 - это рассмотрение примеров. Обучение по примерам эффективно для формирования общего представления о предмете, но мало эффективно для усвоения технических деталей. Из текста учебника можно узнать много технических деталей, но не стоит акцентировать внимание учеников на эти детали. Основное, что хочется, чтобы ученики запомнили - это то, что физически процессор выполняет команды, важность которых видна из разобранных примеров, и, что перед выполнением машины сложной команды на каком-то языке программирования специальная программа (называемая "транслятор") должна эту команду выразить программой, состоящей из последовательности машинных команд.

Хочется обратить внимание учителей на пункт 5 параграфа 2 "Таблица значений". Такое понятие нужно не машине и не конструкторам машин, а людям, которые хотят разобраться в том, что машина делает, как она выполняет программы, написанные в машинных командах.

Нельзя понять программу, только читая ее текст. Обязательно нужно рассматривать, как программа работает.

Таблица значений - это точный аналог таблицы значений, введенной при изучении алгоритмического языка в 9 классе. Она предназначена для единой формальной записи примеров исполнения программ. Можно только сожалеть, что произошли некоторые (хотя и совершенно несущественные) изменения записи. Например, теперь значения аргументов от остальной части таблицы отделяются не прерывистой, а прерывной линией.

Основная трудность для учеников изучения этого материала, по-видимому, будет состоять в том, что они могут не почувствовать главной идеи параграфа. Главное может сместиться с второстепенного, а все запомнить невозможно. Главное - это то, что имеется процессор, имеются регистры (неважно как они обозначаются; тем более, название, который из них используется в качестве счетчика по -

манд). Процессор читает адрес очередной команды, находит и понимает эту команду, читает одно или несколько слов из памяти ЭВМ, записывает эти слова в регистры, выполняет команду (например, сложение) над содержанием регистров (при этом в каком-то из регистров формируется значение результата), пересылает содержание регистра в память ЭВМ.

Остальные подробности этого параграфа ученики забудут довольно скоро.

Домашнее задание: § 2, № 3 § 2.

ОТВЕТЫ НА ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Каков общий вид алгоритма, записанного на алгоритмическом языке?

Ответ. Общий вид алгоритма, записанного на алгоритмическом языке, таков:

алг название алгоритма

нач

команды алгоритма (серия)

кон

2. Какие бывают составные команды? Приведите примеры.

Ответ. В алгоритмическом языке имеются две основные составные команды: команда ветвления и команда повторения (цикла).

Пример команды ветвления. В алгоритме использования международного телефона-автомата всевозможна следующая команда:

если номер нужного телефона содержит меньше чем семи цифр
то перед номером набрать столько цифр "2", чтобы их число вместе с числом цифр в номере телефона было равно семи

все

Пример команды повторения:

пока переключатель каналов телевизора не установлен на "10"

нц

повернуть ручку переключателя каналов на единицу вправо

кц

3. Какие типы величин используются в алгоритмическом языке? Приведите примеры.

Ответ. В алгоритмическом языке используются следующие типы переменных:

нат (натуральный), цел (целый),
вещ (вещественный), лит (литерный).

Примеры переменных типа нат: число шагов, число гирь, делимое, i , j ;
типа цел: делимое, температура;
типа вещ: результат деления, длина отрезка, x , y ;
типа лит: прогноз погоды.

4. Какие величины называются: а) аргументами; б) результатами; в) промежуточными величинами? Приведите примеры.

Ответ. Аргументами называются величины, являющиеся исходными данными алгоритма. Результатами алгоритма называются величины, нахождения значений которых является целью работы алгоритма. Так, в алгоритме нахождения остатка от деления аргументами являются делимое и делитель, а результатом - остаток. Переменная, используемая для обозначения промежуточных значений в процессе выполнения алгоритма, называется промежуточной. Так, при нахождении корней квадратного уравнения промежуточной величиной является дискриминант.

5. Как выполняется команда присваивания?

Ответ. Сначала вычисляется значение выражения, находящегося в правой части команды. Затем это значение объявляется значением переменной, находящейся в левой части команды.

Например, командой присваивания является команда

$D := b^2 - 4 \cdot a \cdot c$. Эта команда читается так: "присвоить переменной D значение выражения $b^2 - 4 \cdot a \cdot c$ "

ВОПРОСЫ И ОТВЕТЫ

1. Что надо сделать, чтобы процессор выполнил заданную программу? Где нужно разместить команды программы? Чему должно равняться содержимое СК?

Ответ. Чтобы процессор выполнил программу, надо команды программы поместить друг за другом в памяти и записать в СК адрес первой команды.

2. Процессор выполнил команду "стоп", расположенную в слове по адресу 1022. Каково после этого содержимое регистра СК?

Ответ. Процессор, работая по основному алгоритму, сначала читает слово из памяти по адресу 1022, затем увеличивает СК на 2, т.е., содержимое СК становится равным 1024, и потом выполняет прочитанную команду "стоп". Поэтому СК после выполнения команды "стоп" будет содержать адрес 1024.

УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Заполните таблицу значений для рассмотренной программы, предполагая, что в начальный момент:

- а) СК=I500, R0= -2I, R1= -I5, R2= -4;
 б) СК=I502, R0=2I, R1= I5, R2=4;
 в) СК=I504, R0=22, R1= I5, R2= 4.

Решение.

а)

Шаг	Команда	СК	R0	R1	R2
		I500	-2I	-I5	- 4
I	переслать слово R0 в R2	I502			-2I
2	добавить слово R1 к R2	I504			-36
3	стоп	I506			

б)

Шаг	Команда	СК	R0	R1	R2
		I502	2I	I5	4
I	добавить слово R1 к R2	I504			I9
2	стоп	I506			

в)

Шаг	Команда	СК	R0	R1	R2
		I504	22	I5	4
I	Стоп	I506			

2. Напишите программу, которая складывает целые числа, находящиеся в регистрах R0, R1, R2, и помещает результат в: а) регистр R3; б) регистр R2.

Решение. Предполагаем, что программа размещается в памяти, начиная с адреса I500.

а)

переслать слово R0 в R3	I500
добавить слово R1 к R3	I502
добавить слово R2 к R3	I504
стоп	I506

б)

добавить слово R0 к R2	I500
добавить слово R1 к R2	I502
стоп.	I504

3. Заполните таблицу значений для программы упражнения 2, считая, что программа размещена начиная с адреса I500, в начальный момент (СК=I500, R0=5, R1=7, R2=8).

Решение.

а) для программы 2а,

Шаг	Команда	СК	R0	R1	R2	R3
		I500	5	7	8	
I	переслать слово R0 в R3	I502			8	5
2	добавить слово R1 к R3	I504				I2
3	добавить слово R2 к R3	I506				20
4	стоп	I508				

б) для программы 2б.

Шаг	Команда	СК	R0	R1	R2
		I500	5	7	8
1	добавить слово R0 к R2	I502			I3
2	добавить слово R1 к R2	I504			20
3	стоп	I506			

4. Напишите программу, которая складывает удвоенное содержимое регистра R0 с утроенным содержимым регистра R1 и помещает результат в:

а) регистр R2; б) регистр R1; в) каждый из регистров R3 и R4.

Решение. Предполагаем, что программа размещается в памяти, начиная с адреса I500.

а)

переслать слово R0 в R2	I500
добавить слово R0 к R2	I502
добавить слово R1 к R2	I504
добавить слово R1 к R2	I506
добавить слово R1 к R2	I508
стоп	I510

Пусть перед началом работы этой программы в R0 лежит число a_0 , а в R1 - число a_1 . Тогда после выполнения первой команды будет $R2 = a_0$, после второй - $R2 = 2 \cdot a_0$, после третьей - $R2 = 2 \cdot a_0 + a_1$, после четвертой - $R2 = 2 \cdot a_0 + 2 \cdot a_1$ и, наконец, после пятой - $R2 = 2 \cdot a_0 + 3 \cdot a_1$.

б) возможны несколько вариантов решений.

Можно использовать программу из пункта 2а, т.е. получить в регистре R2 требуемый результат, а потом его переслать в R1:

переслать слово R0 в R2	I500
добавить слово R0 к R2	I502
добавить слово R1 к R2	I504
добавить слово R1 к R2	I506
добавить слово R1 к R2	I508
переслать слово R2 в R1	I510
стоп	I512

Возможен также более короткий вариант программы:

добавить слово R1 к R0	I500
добавить слово R0 к R1	I502
добавить слово R0 к R1	I504
стоп	I506

Покажем, что после выполнения программы в регистре R1 действительно будет находиться требуемый результат. Пусть в начальный момент $R0 = a_0, R1 = a_1$. После выполнения первой команды в R0 будет число $a_0 + a_1$, в R1 - число a_1 . После выполнения второй команды в R1 будет $(a_0 + a_1) + a_1 = a_0 + 2a_1$. После выполнения третьей команды в R1 будет число $(a_0 + a_1) + (a_0 + 2a_1) = 2a_0 + 3a_1$ т.е. требуемый результат.

в) действуем как в пункте а, только вместо регистра R_2 пользуемся регистром R_3 , и в конце результат из R_3 пересылаем в R_4 :

переслать слово R_0 в R_3	I500
добавить слово R_0 к R_3	I502
добавить слово R_1 к R_3	I504
добавить слово R_1 к R_3	I506
добавить слово R_1 к R_3	I508
переслать слово R_3 в R_4	I510
стоп	I512

Б. Напишите программу, которая помещает в регистр R_1 учетверенное содержание регистра R_0 .

Решение. Предполагаем, что программа размещается в памяти, начиная с адреса I500.

переслать слово R_0 в R_1	I500
добавить слово R_0 к R_1	I502
добавить слово R_0 к R_1	I504
добавить слово R_0 к R_1	I506
стоп	I508

Второй вариант решения:

переслать слово R_0 в R_1	I500
добавить слово R_1 к R_1	I502
добавить слово R_1 к R_1	I504
стоп	I506

Покажем, что после выполнения программы в R_1 будет учетверенное содержание регистра R_0 . Пусть в начальный момент $R_0 = a_0$. После первой команды в R_1 будет a_0 , после второй - $a_0 + a_0 = 2a_0$, после третьей - $2a_0 + 2a_0 = 4a_0$.

ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Напишите программу, которая меняет местами числа, находящиеся в регистрах R_0 , R_1 .

Решение.

переслать слово R_0 в R_2	I500
переслать слово R_1 в R_0	I502
переслать слово R_2 в R_1	I504
стоп	I506

2. Составить таблицу значений для программы из предыдущего упражнения, если $CK = I500$, $R_0 = 2$, $R_1 = 5$.

Решение.

Шаг	Команда	CK	R_0	R_1	R_2
		I500	2	5	
1	переслать слово R_0 в R_2	I502			2
2	переслать слово R_1 в R_0	I504	5		
3	переслать слово R_2 в R_1	I506		2	
4	стоп	I508			

Самостоятельная работа

1. Программа в памяти ЭВМ начинается с адреса I500. После окончания работы программы СК=I506. Сколько команд в программе?

2. Напишите программу, которая помещает в регистры R3 и R5 сумму чисел, лежащих в регистрах R2 и R4.

3. Чему будет равно содержимое регистра R0 после выполнения программы

переслать слово R7 в R0	I500
стоп	I502

4. Для хранения в памяти ЭВМ каждого символа текста требуется один байт; определите, можно ли разместить весь текст учебника "Информатика - IO" в памяти ДВК-2М.

ОТВЕТЫ И РЕШЕНИЯ

1. В программе 3 команды: они хранятся по адресам I500, I502, I504. Последняя из этих команд - команда "стоп". После её исполнения, в соответствии с основным алгоритмом работы процессора, СК=I506.

2. Разместим программу в памяти ЭВМ, начиная с адреса I500:

переслать слово R2 в R3	I500
добавить слово R4 к R3	I502
переслать слово R3 в R5	I504
стоп	I506

3. Ответ: R0=I502. При решении этой задачи нужно не забыть, что R7 - это счётчик команд. Поэтому, согласно основному алгоритму, при исполнении команды "переслать слово R7 в R0" вначале R7 будет увеличено на 2 (до I502), а потом уж полученное число будет переслано в R0.

4. Ответ - не уместится. Действительно, страница содержит больше чем 42 строки, а строка - больше чем 50 букв. Значит, в памяти ДВК-2М помещается меньше чем 28 целиком заполненных страниц.

§ 3. КОМАНДА ВЕТВЛЕНИЯ И КОМАНДА ПОВТОРЕНИЯ (2 ч.)

Название параграфа в учебнике выбрано неудачно: в наборе команд процессора ЭВМ ДВК-2М нет специальных команд ветвления и повторения, а есть другие команды, с помощью которых можно в программах организовать повторение несколько раз одной и той же серии команд или выбор одной из двух серий команд, в зависимости от соблюдения или несоблюдения некоторого условия.

Основная цель. Показать ученикам, каким образом процессор ЭВМ ДВК-2М осуществляет ветвления и повторения работы программы.

Требования к знаниям и умениям. Ученики должны знать команды условного и безусловного перехода. Ученики должны уметь написать простейшие программы (состоящие из одного ветвления или одного повторения) с использованием машинных команд условного и безусловного перехода.

Методические указания

Можно считать, что в результате изучения § 2 учебника познакомились с машинными командами, выполняющими арифметические действия. Однако ограничиться только такими командами нельзя. Дело в том, что составление программ только из арифметических команд имеет серьезный недостаток. Если вам нужно составить программу, которая должна выполнить миллион арифметических действий, то вам в тексте программы придется написать миллион строчек. Это чрезвычайно трудоемко и во многом сводит на нет эффект от быстродействия машины.

Поэтому еще с самого начала развития вычислительных машин было придумано, что кроме арифметических команд можно применять команды другого типа - логические команды, и с их помощью поручить самой программе управление дальнейшим ходом выполнения программы.

В том виде алгоритмического языка, который изучался в 9 классе, логических команд две: команда ветвления и команда повторения.

Команда ветвления дает возможность, в зависимости от того, выполняется ли некоторое логическое условие, выполнять одну или другую серию команд и после этого и в том, и в другом случае перейти к общему продолжению работы.

Применение команды повторения позволяет писать сравнительно короткие программы, которые при исполнении их тем не менее позволяют выполнить очень много арифметических команд (например, немного больше, чем число строчек программы). В тексте программы команда пишется один раз, но, если эта команда находится внутри цикла, то выполняется эта команда может многократно (как правило, всякий раз над другими значениями данных).

В системе команд ЭВМ ДВК-2М вместо команд ветвления и повторения предусмотрены команды условного и безусловного перехода. В § 3 учебника на примерах показано, как команды ветвления и команду повторения реализовать с помощью команд условного и безусловного перехода.

Почему же команды условного и безусловного перехода не включены прямо в алгоритмический язык? Кажется ведь, что включение таких команд дало бы более удобные средства написания программ.

Не имея возможности анализировать здесь все причины, из-за которых возникло так называемое "структурное программирование", отметим, что неограниченное использование команд перехода приводит к столь значительному усложнению синтаксического анализа программ (который нужен, например, при автоматическом переводе с языка программирования высокого уровня на машинные команды или при доказательстве правильности построения программы), что маленькие удобства при написании программы совершенно улетучиваются.

В § 3 учебника на примерах видно, что в рассматриваемых командах переход может совершаться на 1, 2, 3, ... слов, а также на -1, -2, -3, ... слов. В приложении III к учебнику на с. 124 указано, что это число слов (его принято называть смещением) должно находиться в пределах между -128 и +127.

На с. 15 учебника рассматривается пример программы с

повторением. У учеников может возникнуть вопрос, зачем считать сумму $k + (k-1) + \dots + 2 + 1$, когда сразу видно, что эта сумма равна $\frac{k(k+1)}{2}$. Ответ здесь таков: авторы учебника всего лишь хотели привести пример короткой программы, которая выполняется за много шагов. Впрочем, и вычисление по формуле $\frac{k(k+1)}{2}$ не так уж просто. Дело в том, что микропроцессор K1801BM1 не имеет команду умножения. Умножение нужно осуществлять подпрограммой, которая не проще, чем вся программа для вычисления суммы $k + (k-1) + \dots + 2 + 1$.

Домашнее задание: к уроку 1 - § 3 (с. 13-15), № 1 § 3; к уроку 2 - § 3 (с. 15-16), № 3 § 3.

ОТВЕТЫ НА ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Какие знаки отношений между величинами используются в алгоритмическом языке?

Ответ. В алгоритмическом языке используются следующие знаки отношений: < (меньше), > (больше), ≤ (не больше), > (не меньше), = (равно), ≠ (не равно).

2. Какие величины называются табличными? Приведите примеры.

Ответ. Табличными величинами называются величины, представляемые в виде таблиц. Например, таблица умножения, таблица зависимости температуры воздуха от даты измерения.

3. Для чего нужны вспомогательные алгоритмы? Как записывается команда их вызова?

Ответ. Вспомогательные алгоритмы позволяют при создании новых алгоритмов использовать построенные ранее. Если требуется вызвать вспомогательный алгоритм $A(x_1, x_2, \dots, x_n)$, когда x_1, \dots, x_n принимают значения y_1, \dots, y_n , то нужно записать

$$A(y_1, y_2, \dots, y_n).$$

Метод вычисления вспомогательного алгоритма может быть также использован в процессе поиска и разработки нового алгоритма тогда, когда вспомогательный алгоритм еще неизвестен.

ВОПРОСЫ И ОТВЕТЫ

1. Известно, что $R_1 = 5$, $R_2 = 5$, $R_3 = -7$, $R_4 = 0$. Выпишите значения битов N и Z после выполнения команды "справить слово R_1 с R_2 ".

Ответ. После выполнения указанной команды $N=0$, $Z=1$, так как $R_1 - R_2 = 0$.

2. По адресу 1508 расположена команда "переход на +1 слово". Чему будет равно содержимое СК после выполнения этой команды?

Ответ. Процессор выполнит данную команду, если в начале СК = 1508. При выполнении команды действия процессора состоит из четырех этапов: процессор 1) читает адрес из СК; 2) читает слово памяти по этому адресу, т.е. команду "переход на +1 слово"; 3) увеличивает СК на 2, после этого СК = 1510; 4) выполняет команду. Команда "переход на +1 слово" увеличивает СК на 2, поэтому после выполнения команды СК = 1512.

3. В программе 5 команд. Может ли число шагов при выполнении этой программы оказаться равным: а) 1, б) 5, в) 100?

Ответ. Вопрос нечетко сформулирован. Программа не обязательно должна быть одна и та же для пунктов а, б, в.

а) Можно. Если, например, программа состоит из 5 команд "стоп", то выполнится лишь одна команда.

б) Можно, если каждая команда выполняется один раз. Такой будет, например, программа, данная в решении 5 задачи §2.

в) Можно. Рассмотрим программу:

уменьшить R_0 на 1	I500
если меньше, переход на +1 слово	I502
если больше, переход на -3 слово	I504
переслать R_0 в R_1	I506
стоп	I508

Пусть в начальный момент $R_0 = 32$, СК = 1500. Тогда при выполнении этой программы сначала по 32 раза выполнятся первые три команды, при этом содержимое R_0 станет равным 0. После этого выполнятся первая, вторая, четвертая и пятая команды, т.е. число шагов будет $32 \times 3 + 4 = 100$.

УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Напишите программу, после выполнения которой в регистре R_0 будет содержаться: а) 0; б) -1 ; в) наибольшее из чисел, лежащих в R_0, R_1 ; г) наибольшее из чисел, лежащих в R_1, R_2, R_3 .

Решение. Сначала следует согласиться о командах, которых можно использовать при решении. Во-первых, можно использовать все уже знакомые команды. Во-вторых, в командах перехода можно указать переход на $+1, +2, \dots, +127$ и $-1, -2, \dots, -128$ слов (см. приложение 1).

Предполагаем, что программа размещается в памяти, начиная с адреса 1500.

а)

очистить слово R_0	1500
стоп	1502

б)

очистить слово R_0	1500
уменьшить слово R_0 на единицу	1502
стоп	1504

в)

сравнить слово R_1 с R_0	1500
если меньше, переход на $+1$ слова	1502
переслать слово R_1 в R_0	1504
стоп	1506

г) Алгоритм работы программы будет следующий: сначала в R_0 помещается наибольшее из чисел, находящихся в регистрах R_2, R_3 , а потом в регистр R_0 помещается наибольшее из чисел, находящихся в регистрах R_0, R_1 .

На алгоритмическом языке это запишется так (при решении это необязательно писать):

алг большее число (цел R_0, R_1, R_2, R_3)

арг R_1, R_2, R_3

рез R_0

нач

если $R_2 < R_3$

то

$R_0 := R_3$

иначе $R_0 := R_2$

все

если $R_1 < R_0$

то

$R_0 := R_1$

все

кон

Соответствующая программа в памяти:

сравнить слово R_2 с R_3	1500
если меньше, переход на $+2$ слова	1502
переслать слово R_2 в R_0	1504
переход на $+1$ слово	1506
переслать слово R_3 в R_0	1508
сравнить слово R_1 с R_0	1510
если меньше, переход на $+1$ слово	1512
переслать слово R_1 в R_0	1514
стоп	1516

2. Напишите программу, после выполнения которой в регистре $R0$ будет содержаться сумма $2k+(2k-2)+\dots+4+2$, если перед началом работы число k хранится в регистре $R1$.

Решение. Программа вначале удваивает содержимое регистра $R1$, и в дальнейшей работе программы $R1$ будет содержать очередное слагаемое: $2k, 2k-2, \dots, 4, 2$. Когда $R1$ станет равным нулю, выполнение программы закончится. $R2$ содержит -2 , добавляя которую к регистру $R1$, получается очередное слагаемое.

Расположим соответствующую программу в памяти ЭЕМ, начиная с адреса I500.

очистить слово $R0$	I500
добавить слово $R1$ к $R1$	I502
очистить слово $R2$	I504
уменьшить слово $R2$ на единицу	I506
уменьшить слово $R2$ на единицу	I508
добавить слово $R1$ к $R0$	I510
добавить слово $R2$ к $R1$	I512
если больше, переход на -3 слова	I514
стоп	I516

3. Напишите программу, после выполнения которой в регистре $R0$ будет содержаться $999+997+\dots+3+1$. Перед началом работы $R1 = 500$.

Решение.

очистить слово $R0$	I500
добавить слово $R1$ к $R1$	I502
уменьшить слово $R1$ на единицу	I504
очистить слово $R2$	I506
уменьшить слово $R2$ на единицу	I508
уменьшить слово $R2$ на единицу	I510
добавить слово $R1$ к $R0$	I512
добавить слово $R2$ к $R1$	I514
если больше, переход на -3 слова	I516
стоп	I518

ДОПОЛНИТЕЛЬНЫЕ УПРАЖНЕНИЯ И РЕШЕНИЯ

I. Напишите программу, после выполнения которой в регистре $R0$ будет содержаться результат умножения двух положительных чисел, находящихся в регистрах $R1$ и $R2$.

Решение. Умножение выполняется по следующему алгоритму:

алг умножение (цел $R0, R1, R2$)

арг $R1, R2$

рез $R0$

нач

$R0 := 0$

пока $R1 \neq 0$

нц

$R0 := R0 + R2$

$R1 := R1 - 1$

кц

кон

Соответствующая программа на машинном языке будет выглядеть так:

очистить слово R0	I500
добавить слово R2 к R0	I502
уменьшить слово R1 на единицу	I504
если больше, переход на -3 слова	I506
стоп	I508

2. Заполнить таблицу значений для программы из упражнения 2 учебного пособия, считая, что в начальный момент СК=I500, R1 =2.

Решение.

Шаг	Команда	СК	R0	R1	R2	N	Z
		I500		2			
I	очистить слово R0	I502	0			0	I
2	добавить слово R1 к R1	I504		4		0	0
3	очистить слово R2	I506			0	0	I
4	уменьшить слово R2 на единицу	I508			-1	I	0
5	уменьшить слово R2 на единицу	I510			-2	I	0
6	добавить слово R1 к R0	I512	4			0	0
7	добавить слово R2 к R1	I514		2		0	0
8	если больше, переход на -3 слова	I510					
9	добавить слово R1 к R0	I512	6			0	0
10	добавить слово R2 к R1	I514		0		0	I
11	если больше, переход на -3 слова	I516					
12	стоп	I518					

§4. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ЭВМ (I ч.)

Основная цель. Не вдаваясь в технические детали, дать ученикам самое общее представление о кодировке в ЭВМ адресов, чисел, символов и команд с помощью двоичных кодов.

Требования к знаниям и умениям. Ученики должны знать, что существует несложная кодировка в ЭВМ адресов, чисел, символов и команд.

Методические указания

Материал этого параграфа был бы легким и почти очевидным, если бы ученики знали двоичную систему счисления. Однако они это не проходили, и не стоит им это излагать только из-за материала этого урока, конкретные детали которого в дальнейшем курсе уже больше не понадобятся.

Ученики должны понять, что байт (состоящий из 8 битов) может принимать 2^8 различных значений. Это лучше всего продемонстрировать, показав, что последовательность, состоящая из 3 битов, может принимать $2^3 = 8$ значений. Желательно все эти 8 возможных значений написать на доске:

```

000
001
010
011
100
101
110
111

```

Потом обратить внимание учеников на регулярность расположения единиц и нулей: сначала в первом столбце идут нули, а в это время в последних 2 столбцах встречаются все мыслимые расположения нулей и единиц. Потом - в первом столбце все единицы, а в последних 2 столбцах опять те же самые расположения нулей и единиц.

Ученики должны сами додуматься до того, что последовательность, состоящая из 4 битов, может принимать в 2 раза больше значений, а именно 8 таких значений, где в первом столбце нуль, и потом 8 таких, где в пер-

вом столбце единица.

Следующий шаг рассуждения уже показывает, что последовательность, состоящая из n битов, может принимать 2^n различных значений. Следовательно, байт может принимать $2^8 = 256$ различных значений, а машинное слово — $2^{16} = 65536$ различных значений.

У учителя может возникнуть несколько естественных вопросов.

Вопрос 1. Нужно ли для понимания кодировки знать правила арифметических действий в двоичной системе счисления?

Ответ. Для понимания работы процессора в объеме, предусмотренном программой и учебником, достаточно понимать, что каждому адресу от 0 до 65535 взаимно однозначно сопоставлен двоичный код, и каждому целому числу от -32768 до +32767 также взаимно однозначно сопоставлен двоичный код.

Вопрос 2. Один и тот же двоичный код может означать и адрес, и число, и команду. Не может ли это вызвать путаницу?

Ответ. Нет, не может. Путаница могла бы возникнуть, если бы процессор получил какой-то код и должен был выяснить, что это такое и что с ним делать. На самом деле процессор всегда получает код вместе с информацией, что с этим кодом делать. Например, при выборе команды содержимое СК всегда расшифровывается как адрес, слово по этому адресу расшифровывается как команда, а если это команда "добавить слово $R1$ к слову $R2$ ", то $R1$ и $R2$ расшифровывается как числа.

Вопрос 3. Можно ли на ЭВМ ДВК-2М работать с числами, которые больше чем 32767?

Ответ. Можно, но тогда программист должен сам придумать свой нестандартный способ кодирования чисел. Такой код будет занимать в памяти ЭВМ места больше чем одно машинное слово. При новом способе кодирования сложение (и другие операции над числами) нельзя осуществить одной машинной командой. Придется написать специальную подпрограмму для каждой арифметической операции, да и исполнение этой подпрограммы займет больше времени, чем выполнение одной машинной команды при стандартном кодировании чисел. Это затруднительно, но сделать можно.

Д о м а ш н е е з а д а н и е: §4, №1, 3 §4.

ОТВЕТЫ НА ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ

1. Какие вы знаете команды работы с графической информацией? Приведите примеры алгоритмов, использующих эти команды.

Ответ. Команды работы с графической информацией: вперед (а), назад (в), налево (в), рисуй, не рисуй. Примеры: алгоритмы написания слова МИР, рисования спирали, квадрата.

ВОПРОСЫ И ОТВЕТЫ

1. Каково число различных последовательностей из трех цифр, каждое из которых есть либо ноль, либо единица?

Ответ. Так как любая из трех цифр может принимать два значения — ноль или единицу, то имеется $2 \cdot 2 \cdot 2 = 8$ различных последовательностей.

УПРАЖНЕНИЯ И РЕШЕНИЯ

1. Какой адрес кодируется последовательностью
1 000 000 000 000 000 ?

Решение.
Адрес = $2^{15} \cdot 1 + 2^{14} \cdot 0 + \dots + 2^1 \cdot 0 + 2^0 \cdot 0 = 2^{15} = 1024 \cdot 32 = 32768$.

2. Какой последовательностью кодируется адрес 1024?

Решение.
 $1024 = 2^{10} = 0 \cdot 2^{15} + 0 \cdot 2^{14} + \dots + 1 \cdot 2^{10} + 0 \cdot 2^9 + \dots + 0 \cdot 2^0$, следовательно, адрес 1024 кодируется последовательностью
0000010000000000.

3. Какое целое число кодируется последовательностью

1 000 000 000 000 000 ?

Решение. Используем соответствие между числом и его кодом:
число = $i_{15} \cdot 2^{15} + i_{14} \cdot 2^{14} + \dots + i_0 \cdot 2^0 = i_{15} \cdot 2^{16} =$
 $= 1 \cdot 2^{15} - 1 \cdot 2^{16} = 32768 - 65536 = -32768$.

§ 5. ФИЗИЧЕСКИЕ ПРИНЦИПЫ РАБОТЫ ЭВМ

СОДЕРЖАНИЕ МАТЕРИАЛА

Материал § 5 учебника содержит следующие вопросы:

1. Применение электрических сигналов для кодирования двоичной информации в ЭВМ.
2. Простейшие электронные устройства для обработки информации. Элементы "НЕ", "И-НЕ".
3. Устройства и физические принципы работы клавиатуры, дисплея, принтера и накопителя на гибких магнитных дисках.

ОСНОВНЫЕ ЦЕЛИ

1. Ознакомить учащихся с физическими принципами преобразования двоичной информации в микроЭВМ с использованием электронных схем, построенных на элементарных логических ячейках.
2. Рассмотреть конструкцию и физические принципы работы основных узлов устройств ввода/вывода микроЭВМ: клавиатуры, дисплея, печатающего устройства.
3. Рассмотреть конструкцию и физические принципы работы основных узлов устройств внешней памяти микроЭВМ: магнитофона и накопителя на гибких магнитных дисках.
4. Повторить и развить материал отдельных параграфов курса физики 9 и 10 классов (см. [1, § 76, 81, 89, 90, 93, 93, 98] и [2, § 12, 50, 80] из списка литературы к § 5 настоящего пособия).

ТРЕБОВАНИЯ К ЗНАНИЯМ И УМЕНИЯМ

Учащиеся должны разбираться в физических принципах представления и преобразования двоичной информации в микроЭВМ с применением электронных логических схем, иметь представление о конструкции основных узлов и физических принципах работы периферийных устройств микроЭВМ: клавиатуры, принтера, дисплея, накопителя на гибких магнитных дисках.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Приступая к изучению темы, необходимо сделать обзор сведений об устройстве микроЭВМ и периферийных устройств, опираясь на знания учащихся, полученные при изучении предыдущих параграфов, а также на опыт общения с бытовой электронной техникой. Следует подчеркнуть, что некоторые выпускаемые промышленностью микроЭВМ (например, "Электроника ЕК-0010", "Агат") комплектуются бытовым телевизором и магнитофоном.

Изучение темы целесообразно сопровождать демонстрацией различных узлов микроЭВМ и периферийных устройств (в том числе в процессе работы). При отсутствии микроЭВМ в классе желательно провести экскурсию в вычислительный центр, оснащенный микропроцессорной техникой или приурочить первое за-

нятие по теме к передаче по учебному телевидению.

Поскольку по теме проведение практических занятий не запланировано, материал рекомендуется излагать в форме беседы.

1. Логические схемы и преобразование информации

Содержанием первого рассматриваемого на уроке вопроса является кодирование двоичной информации и ее обработка с применением электронных устройств.

Следует обратить внимание учащихся на то, что электронные схемы способны различать два уровня напряжений - низкий (от 0 до 0,5 В) и высокий (от 2,5 до 5 В) - при напряжении питания схем 5 В. Эти уровни напряжений отождествляются соответственно с сигналами "1" и "0". Для обработки информации в ЭВМ применяются электронные схемы, преобразующие по заданным законам сигналы "1" и "0".

В основе преобразования информации лежит разработанная английским математиком Булем алгебра логики, изучающая действия (логическое сложение, умножение и др.) над величинами, имеющими только два значения "ДА" и "НЕТ".

Взаимосвязь принципов работы электронных схем, обрабатывающих двоичную информацию, состоящую из совокупности "1" и "0", в булевой алгеброй логики была установлена американским математиком Нейманом. Позже Шенноном, Шестаковым и Нейманом было доказано, что все основные электронные схемы ЭВМ могут быть составлены из трех базовых элементов, реализующих логические операции "И", "ИЛИ" и "НЕ": другие логические элементы (например, "И-НЕ") являются сочетаниями базовых элементов (см. ниже).

Следует перечислить действия, выполняемые с помощью электронных логических схем: запоминание, преобразование и пересылка информации, включение и отключение периферийных устройств и другие, - а также подчеркнуть, что управляет этими действиями микропроцессор по соответствующим программам.

Для более подробного ознакомления с алгеброй логики и разнообразными логическими схемами можно рекомендовать статьи [5 - 11] и отдельные параграфы книги [21].

Элементы "И", "ИЛИ", "НЕ"

Комментарии к тексту 2-го и 3-го абзацев на стр. 20, 1-го и 2-го абзацев на стр. 21.

В тексте учебника рассказано о работе инвертора (логического элемента "НЕ") и логического элемента "И-НЕ" (см. рис. и табл. 5.1 и 5.4 и ответы на вопросы № 2 и 3 учебника, приведенные в настоящих методических указаниях). Далее рассмотрим работу логических элементов "ИЛИ" и "И".

Элемент "ИЛИ" - это электронная схема с одним выходом, а также двумя и более входами (рис. 5.2). На выходе элемента "ИЛИ" появляется низкое напряжение ("1"), если хотя бы на один из входов подано высокое напряжение ("1"). На выходе напряжение низкое ("0"), если на всех входах напряжения низ-

Рис. и табл. 5.1. Логическая схема "НЕ".

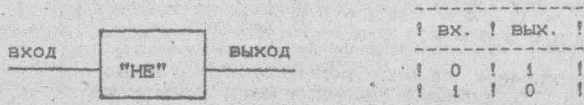


Рис. и табл. 5.2. Логическая схема "ИЛИ".

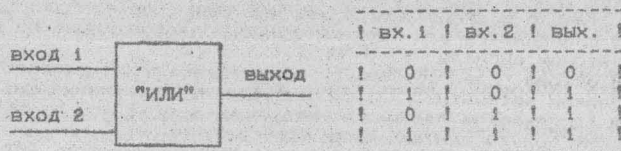


Рис. и табл. 5.3. Логическая схема "И".

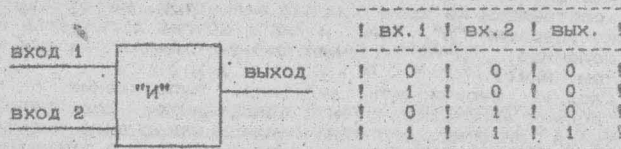
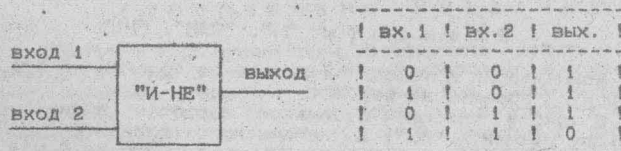


Рис. и табл. 5.4. Логическая схема "И-НЕ".



кое ("0"). Работа элемента "ИЛИ" с двумя входами задается табл. 5.2.

Схемы, реализующие элемент "ИЛИ", называют также схемами логического сложения: результат логического сложения (логическая сумма) равен единице, если хотя бы одно из слагаемых равно единице.

Элемент "И" - это электронная схема с одним выходом, с также двумя и более входами (рис. 5.3). На выходе элемента "И" появляется высокое напряжение ("1") только в том случае, если на все входы ячейки подается высокое напряжение ("1"). Если хотя бы на одном из входов напряжение низкое ("0"), то и на выходе напряжение низкое ("0"). Работа элемента "И" с двумя входами задается табл. 5.3.

Схемы, реализующие элемент "И", называются также схемами логического умножения: если хотя бы один из сомножителей при логическом умножении равен нулю, то и значение произведения равно нулю.

Легко проверить, что логический элемент "И-НЕ" (рис. 5.4) получается последовательным соединением логических элементов "И" (рис. 5.3) и "НЕ" (рис. 5.1). Информация преобразуется в соответствии с табл. 5.4.

Элементарная ячейка памяти - триггер

В дополнение к материалу учебника приведем пример сложной логической схемы - триггера, сочетающей по два логических элемента "ИЛИ" и "НЕ" (рис. 5.5). Триггер предназначен для запоминания информации.

1. Работа триггера в режиме запоминания информации.

Пусть первоначально на обоих входах и выходах установлен сигнал "0". Для записи информации подадим на "вход 1" сигнал "1", тогда на "выходе 2" установится сигнал "1", не зависящий от сигнала на "входе 2" ("вход 2" подключен к логическому элементу "ИЛИ", см. табл. 5.2). Если теперь подать сигнал "0" на "вход 1", то сигналы на "выходе 1" и "выходе 2" не изменятся (ср. рис. 5.5-а и 5.5-б).

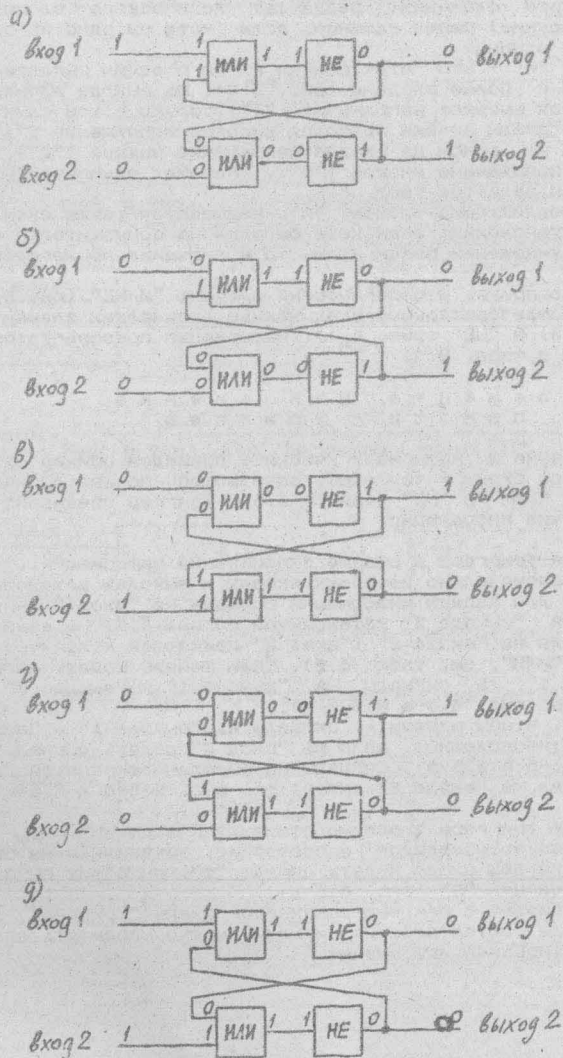
Для того, чтобы изменить сигналы на "выходе 1" и "выходе 2" на противоположные, надо на "вход 2" подать сигнал "1" (ср. рис. 5.5-б и 5.5-в). Сигналы на выходах сохранятся, даже если сигнал на "входе 2" снять (ср. рис. 5.5-в и 5.5-г).

2. Работа триггера в режиме стирания информации.

Пусть триггер находится в состоянии, показанном на рис. 5.5-б. Если на оба входа подать сигнал "1", на обоих выходах установится сигнал "0" (рис. 5.5-д).

Таким образом, с помощью логической схемы "триггер", подавая на входы соответствующие сигналы, можно управлять запоминанием и стиранием информации.

Рис. 5.5. Логическая схема триггер.



2. Реализация логических схем на транзисторах технология изготовления электронных схем

Комментарий к тексту 3-го абзаца стр. 21.

Из введения и § 1 учебника учащиеся знают, что электронные схемы ЭВМ, реализующие разнообразные сочетания логических операций, состоят из большого числа полупроводниковых элементов, объединенных в интегральные схемы. Интегральные схемы изготавливаются по специальной т.н. планарной технологии на одном кристалле полупроводника. В тексте § 5 учебника об этом только упоминается, однако при изучении физических принципов работы ЭВМ перечисленные вопросы должны быть рассмотрены достаточно подробно (отметим, что они присутствуют в программе курса). Рекомендуется их включить в курс физики 8-го класса как дополнение к изучаемому материалу по полупроводникам или же вынести его на факультативные занятия по физике или основам информатики и вычислительной техники, что придаст им необходимую прикладную направленность.

Перечислим темы, требующие изучения:

- принципы работы биполярных и полевых транзисторов;
- реализация базовых логических элементов в виде транзисторных схем и их работа;
- транзисторные схемы, реализующие запоминание и преобразование информации, и их работа;
- технология изготовления транзисторов в интегральных схемах.

Для подготовки к урокам по перечисленным темам рекомендуется использовать статьи [8 - 12], ориентированные на учащихся, а также отдельные параграфы книги [21]. Дополнительный материал, касающийся терминологии, можно почерпнуть в словарях [18 - 20].

3. Периферийные устройства микроЭВМ

При изучении физических принципов работы периферийных устройств компьютера изложение рекомендуется вести по тексту учебника, где материал изложен достаточно подробно (за исключением клавиатуры). В качестве дополнительной информации учителю могут быть полезными следующие комментарии.

Клавиатура компьютера

Комментарий к тексту 5-го и 6-го абзацев на стр. 21 (см. также ответ на вопрос # 4).

Можно уточнить, как электронная схема "узнает", какая именно клавиша нажата.

Все контакты клавиатуры образуют некоторую упорядоченную структуру. В большинстве клавиатур переключатели объединены в матрицу (рис. 5.6). Матрица состоит из вертикальных и го-

Рис. 5.6. Электрическая схема переключателей клавиатуры компьютера.

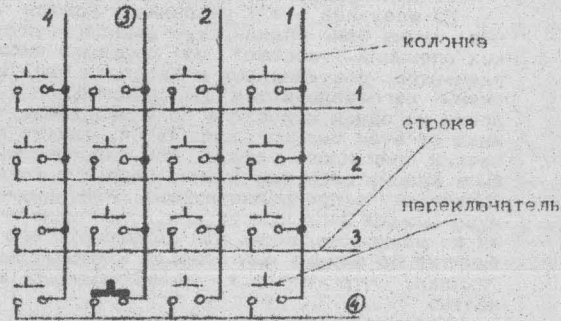
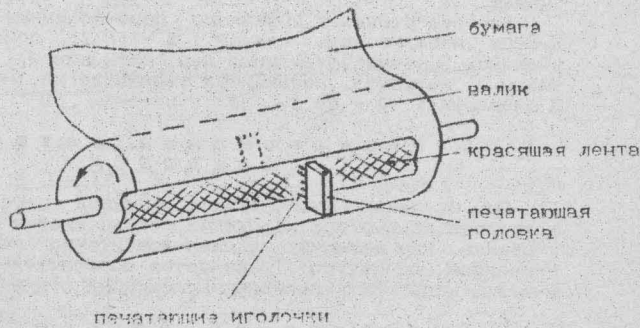


Рис. 5.7. Схема печатающего узла принтера.



ризонгальных проводов-линий (колонок и строк). На каждом из пересечений колонок и строк имеется переключатель, замыкающий накоротко колонку и строку, соответствующие нажатой клавише (на рис. 5.6 при нажатом переключателе (выделен жирным) замыкается 3-я колонка и 4-я строка).

Специальная "сканирующая" схема, соединенная с колонками, посылает сигналы на 1-ю, 2-ю, 3-ю, 4-ю колонки, затем повторяет сигналы в той же последовательности. При нажатии клавиши переключатель замыкает цепь, и в схему - "анализатор", подключенную к строкам, поступает сигнал, на пересечении какой колонки и строки замкнут контакт.

Слабым местом клавиатуры является переключатель, поэтому "конструкторы ЭВМ постоянно придумывают" новые конструкции переключателей, которые при нажатии клавиши обеспечивают стабильный сигнал.

В соответствии с полученным сигналом электронная схема формирует 8-битный код символа, набираемого на клавиатуре, и этот код обрабатывается процессором.

Следует акцентировать внимание учащихся на том, что функция клавиатуры состоит в формировании 8-битного кода при нажатии клавиши. Анализ кода производится процессором по соответствующим программам. Нажатие клавиши может восприниматься как:

- ввод алфавитно-цифрового или специального символа;
- выбор прописной или строчной буквы;
- выбор алфавита (русского, латинского, национального);
- команда по перемещению курсора;
- ввод строки символьной информации в ЭВМ;
- прерывание работы программы.

При необходимости дополнительную информацию можно найти в книге [22].

Печатающее устройство

Комментарий к тексту 7-го абзаца на стр. 21 и 1-го и 2-го абзацев на стр. 23 (см. также ответ на вопрос # 5).

В учебнике рассказывается о матричном принтере, в котором изображение на бумаге формируется из совокупности точек. Рекомендуется уточнить, как работают при выводе информации на бумагу электромеханические узлы принтера.

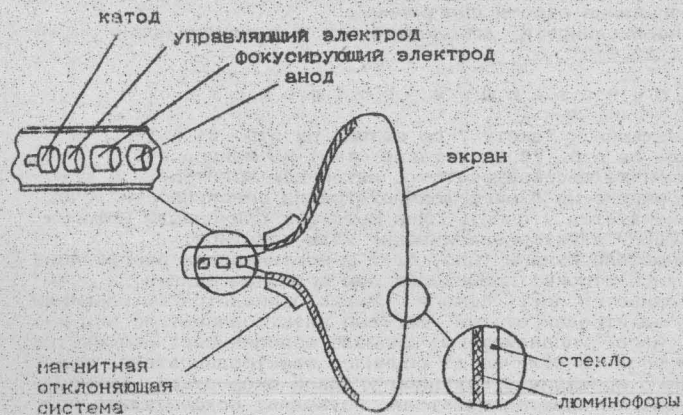
Перемещение бумаги (рис. 5.7) в вертикальном направлении (от строки к строке) происходит при вращении валика, к которому прижимается лист бумаги. Валик приводится во вращение шаговым электродвигателем. Шаговый электродвигатель при подаче на него управляющего сигнала обеспечивает поворот на заданный угол (в отличие от обычного электродвигателя, обеспечивающего вращение с заданной угловой скоростью). Применение шагового электродвигателя позволяет осуществлять программно-управляемое перемещение бумаги.

Совокупность точек, из которых создается изображение, выводится на бумагу посредством печатающей головки (рис. 5.8). Печатающая головка состоит из нескольких (около десяти) вертикально расположенных иглолочек. Каждая иглолочка при

Рис. 5.8. Схема печатающей головки.



Рис. 5.8. Схема кинескопа дисплея цветного изображения.



водится в движение своим миниатюрным электромагнитом. При пропускании тока через обмотку электромагнита иглолочка выталкивается и наносит удар по красящей ленте и таким образом оставляет след на бумаге. Сигналы электромагнитам подаются блоком управления печатающей головки.

В горизонтальном направлении печатающая головка перемещается посредством механизма позиционирования с приводом от шагового электродвигателя. Применение последнего позволяет программно управлять расстоянием (порядка половины толщины иглолочки), на которое передвинется головка относительно бумаги.

Принтер работает следующим образом: одновременно на бумагу выводится вертикальный ряд точек, затем с помощью механизма позиционирования головка перемещается в горизонтальном направлении на заданное расстояние, и на бумагу выводится еще один вертикальный ряд точек до тех пор, пока не будет напечатана строка информации. Затем лист бумаги перемещается в вертикальном направлении на заданное расстояние (при выводе текста между строками остается свободное место, при выводе рисунка промежутки отсутствуют).

При выводе текста на матричном принтере может производиться программно-управляемая смена шрифтов. Каждый символ изображается совокупностью точек, расположенных на некоторой сетке-матрице: например, 8×7 . В памяти принтера хранится несколько вариантов шрифтов (прописные и строчные буквы русского, латинского и других алфавитов). Принтер выводит на печать символ после получения от компьютера соответствующего символу 8-битного сигнала. Устройство управления принтера по матрице символа, хранящейся в памяти формирует управляющие сигналы для исполнительных механизмов.

Для вывода информации из ЭВМ применяются также принтеры других конструкций. Существуют устройства вывода информации на фотопленку и фотобумагу (см. [17]).

Д и с п л е й к о м п ь ю т е р а

Комментарий к тексту 3-го абзаца на стр. 28 (см. также ответ на вопрос # 8).

При изложении материала следует подчеркнуть, что в учебнике изложен принцип действия черно-белого дисплея.

Целесообразно систематизировать и расширить материал об устройстве электронно-лучевой трубки из курса физики 9-го и 10-го классов [1, 2] (см. также рис. 5.9):

- устройство электронной пушки, состоящей из катода, анода, управляющего и фокусирующего электродов;

- устройство и принцип действия магнитной отклоняющей системы электронного луча (ранее рассматривалась электростатическая отклоняющая система);

- принцип развертки электронного луча и управление его энергией;

- реакция люминофорного покрытия, нанесенного на внутреннюю поверхность экрана, на воздействие электронного луча.

У учащихся, знакомых с устройством телевизора, может

возникнуть вопрос - отличается ли развертка электронного луча в дисплее и телевизоре. В дисплее развертка построчная, а в телевизоре развертка чересстрочная (сначала луч пробегает все нечетные строчки, а затем все четные строчки).

Можно рассказать также о принципе действия цветного телевизора. Для получения изображения на экране, необходимо управлять тремя электронными лучами. Попадая на внутреннюю поверхность экрана, каждый луч возбуждает свечение люминофора определенного цвета. Одновременное свечение люминофоров трех различных цветов создает впечатление цвета, образуясь при смешении этих цветов. Чтобы воспрепятствовать попаданию электронного луча на область экрана, покрытую люминофором, дающим свечение другого, не соответствующего лучу, цвета, перед экраном располагается специальная теньевая маска (рис. 5.10).

Следует также пояснить учащимся, что поскольку изображение символов на экране дисплея формируется из точек также, как и в принтере, может быть создано несколько вариантов шрифтов с возможностью их программно-управляемого выбора.

Отметим также, что начертание символов, воспроизводимых на экране дисплея и печатаемых с помощью принтера, могут не совпадать, поскольку в микроЭВМ таблицы шрифтов для экрана и принтера формируются независимо.

Дополнительная информация содержится в статьях [13-16, 18, 20].

4. Устройства внешней памяти

Для долговременного хранения информации к компьютеру подключаются внешние запоминающие устройства, к которым относятся магнитофон и накопитель на гибких магнитных дисках.

М А Г Н И Т О Ф О Н

Комментарий к тексту 2-го абзаца на стр. 24.

В учебнике при изложении материала есть ссылка на принцип работы магнитной головки магнитофона, однако в курсе физики 9-го класса этот вопрос не рассматривается.

Поясним принцип работы магнитной головки магнитофона (рис. 5.11). Магнитная головка состоит из сердечника, изготовленного из "мягкого" ферромагнитного материала, согнутого в виде подковы с маленьким зазором, обращенным к магнитной ленте, имеющей покрытие из "жесткого" ферромагнитного материала. Вокруг сердечника намотана обмотка.

При пропускании тока по обмотке в режиме записи магнитный поток в сердечнике меняется в соответствии с подаваемым сигналом. Поскольку покрытие магнитной ленты в области зазора включается в магнитную цепь, в нем происходит переориентация областей спонтанной намагниченности ("доменов"), и сохраняется информация о сигнале в обмотке сердечника.

В режиме чтения магнитная лента с записанной информацией проходит мимо головки. При этом индуцируется магнитный поток, пронизывающий сердечник, в обмотке которого по закону Фара-

Рис. 5.10. Теньевая маска кинескопа дисплея.

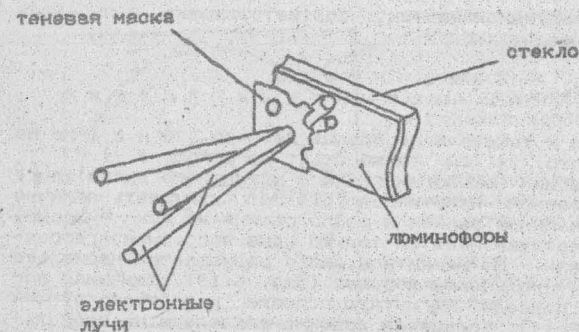
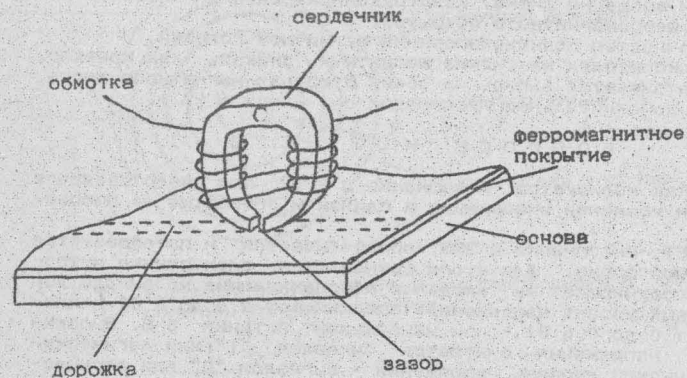


Рис. 5.11. Схема магнитной головки.



дея наводится индукционный ток, соответствующий записанной информации.

Накопитель на гибких магнитных дисках

Комментарий к тексту 4-го абзаца на стр. 28 и с 1-го по 5-ый абзац на стр. 24 (см. также ответ на вопрос # 7).

При проведении аналогии гибкого магнитного диска (рис. 5.12) с граммофонной пластинкой следует учитывать отличие физических принципов записи и расположения на них информации. На грампластинке имеется только одна непрерывная дорожка в виде спирали. На магнитном диске размещается несколько десятков (30-80) кольцевых дорожек (рис. 5.13). Механизм позиционирования подводит магнитную головку к соответствующей дорожке (рис. 5.14). Кольцевая дорожка разделена на несколько секторов, в пределах каждого из которых записывается несколько сотен единиц информации (например, 256 байт).

Количество дорожек, а также количество секторов в дорожке определяется:

- физической возможной плотностью записи при заданных скорости вращения диска, конструкции магнитной головки и материале ферромагнитного покрытия;
- точностью позиционирования магнитной головки.

В накопителях на гибких магнитных дисках, как правило, предусматривается запись на обеих сторонах магнитного диска. Для дополнительной информации см. книгу [22].

ВОПРОСЫ И ОТВЕТЫ

1. Как кодируется информация в ЭВМ? Как представляются биты при хранении информации в памяти и обработке ее процессором?

ОТВЕТ. Информация в ЭВМ обрабатывается и преобразуется в двоичной форме. Для этого применяются электронные микросхемы, получающие на входе и обеспечивающие на выходе два устойчивых уровня напряжения: высокий (от 2,5 до 5 В) и низкий (от 0 до 0,5 В) - при напряжении питания 5 В. Высокий уровень напряжения считается сигналом "1" или логическим "ДА", низкий уровень напряжения - сигналом "0" или логическим "НЕТ".

2. Что такое инвертор?

ОТВЕТ. Инвертор - это электронная схема с одним входом и одним выходом (рис. 5.1), преобразующая информацию в соответствии с табл. 5.1:

- если на вход схемы подается высокое напряжение ("1" или "ДА"), то на выходе появляется низкое напряжение ("0" или "НЕТ");
- если на вход схемы подается низкое напряжение ("0" или "НЕТ"), то на выходе появляется высокое напряжение ("1" или "ДА").

Сигнал на выходе инвертора "отрицает" сигнал на входе

Рис. 5.12. Устройство гибкого магнитного диска.

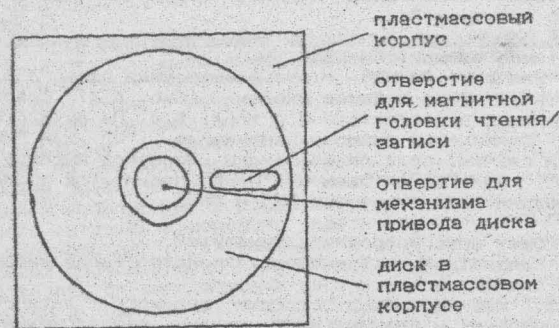


Рис. 5.13. Расположение информации на магнитном диске.

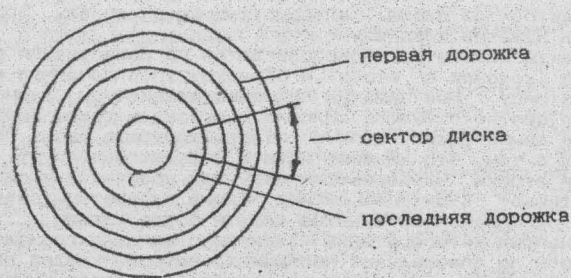
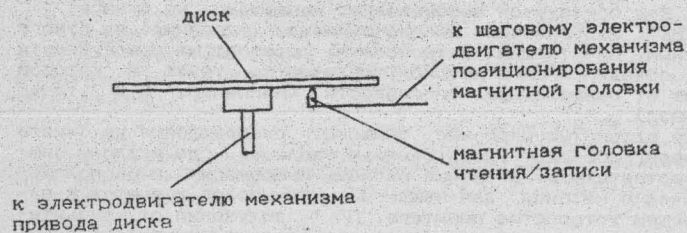


Рис. 5.14. Взаимное расположение магнитного диска и головки чтения / записи.



(сигнал "0" означает "НЕ-1" и наоборот), поэтому инвертор называется также логической ячейкой типа "НЕ" или схемой "логического отрицания".

3. Что такое элемент "И-НЕ"?

ОТВЕТ. Элемент "И-НЕ" - это электронная схема с одним выходом, а также двумя и более входами (рис. 5.4), преобразующая информацию в соответствии с табл. 5.4. На выходе элемента "И-НЕ" появляется низкое напряжение ("0" или "НЕТ") только в том случае, если на все входы подается высокое напряжение ("1" или "ДА"). Элемент "И-НЕ" получается последовательным соединением элементов "И" и "НЕ".

4. Как может быть устроена клавиатура?

ОТВЕТ. Клавиатура - клавишное устройство ввода информации в ЭВМ - состоит из:

- клавиш, маркированных буквами русского и латинского алфавитов, цифрами и специальными знаками, расположение которых соответствует пишущей машинке (исключая некоторые специальные клавиши, отсутствующие в пишущей машинке);
- матрицы переключателей, которые замыкаются при нажатии клавиш;
- электронной схемы, интерпретирующей сигналы, образующиеся при нажатии клавиш.

Контакты клавиатуры располагаются на пересечении проводов-линий (колонок и строк) и образуют упорядоченную структуру - матрицу. При нажатии клавиши клавиатуры замыкается накоротко соответствующая колонка и строка матрицы проводов-линий. По замкнувшейся цепи в интерпретирующую схему подается сигнал о том, что клавиша нажата. Схема фиксирует, номер колонки и строки, образовавших цепь для прохождения сигнала, и вырабатывает 8-битный сигнал. Схема клавиатуры реагирует также на одновременное нажатие двух и более клавиш.

Дальнейшая интерпретация 8-битного сигнала производится процессором по специальной программе, которая каждой нажатой клавише ставит в соответствие алфавитно-цифровой символ или функцию, управляющую работой ЭВМ или находящейся в ней программы. Общее число таких символов и функций дается количеством различных комбинаций, обеспечиваемых 8-битным сигналом, и равно 256.

5. Как образуется изображение, печатаемое на бумаге?

ОТВЕТ. Изображение любого символа, выводимое на бумагу матричным печатающим устройством, состоит из совокупности точек, объединенных в матрицу, количество точек в которой зависит от разрешающей способности (например, 9 x 7 точек на символ).

Все алфавитно-цифровые символы, отображаемые на бумаге принтером, образуют упорядоченную таблицу, где каждому символу соответствует 8-битный сигнал. Информация о расположении точек в матрице для каждого из символов хранится в запоминающем устройстве принтера. После получения от компьютера 8-битного сигнала устройство управления принтера в соот-

ветствии с записанной в памяти информацией о начертании символа, передает управляющие сигналы на исполнительные механизмы, к которым относятся: печатающая головка; шаговый двигатель позиционирования головки; шаговый двигатель перемещения валика.

В памяти принтера может храниться несколько вариантов шрифтов одного или нескольких алфавитов (русского, латинского и национального), что дает возможность программно управлять выбором языка и начертания шрифта для выводимой алфавитно-цифровой информации.

Изображение переносится на бумагу печатающей головкой, состоящей из вертикально расположенных иглоочек. Устройством управления подается сигнал на электромагниты, управляющие переключением иглоочек в перпендикулярном к бумаге направлении. На бумаге остается комбинация вертикально расположенных точек в соответствии с матрицей воспроизводимого символа. Так как символ состоит из нескольких групп вертикально расположенных точек, он печатается в несколько приемов после сдвига головки в горизонтальном направлении на расстояние, равное толщине иглы.

В горизонтальном направлении головка перемещается механизмом позиционирования с приводом от шагового электродвигателя. Вертикальное перемещение листа бумаги происходит при вращении валика, также приводимого в движение шаговым электродвигателем. Применение шаговых двигателей позволяет программно управлять движением листа бумаги и печатающей головкой.

6. Как образуется изображение на экране дисплея?

ОТВЕТ. Изображение на экране дисплея аналогично принтеру состоит из точек. На экране дисплея появляется светящаяся точка, если на нее падает электронный луч. Яркость точки на экране зависит от величины потенциала, подаваемого на управляющий электрод электронно-лучевой трубки дисплея: потенциал можно установить таким, что энергии электронного луча не будет достаточно для возбуждения свечения люминофорного покрытия внутренней поверхности экрана.

Разверткой электронного луча управляет магнитная отклоняющая система, подводящая луч при поступочном обходе последовательно ко всем точкам на экране (в телевизоре развертка черезстрочная). При движении электронного луча вдоль строки управляющий электрод подается сигнал, зависимость величины которого от времени определяется необходимым расположением вдоль строки светящихся точек различной яркости.

Сидящий за дисплеем воспринимает на экране устойчивую светящуюся картинку ввиду большой скорости развертки и незначительного послесвечения люминофора на внутренней поверхности экрана.

7. Как записывается информация на магнитном диске?

ОТВЕТ. Для записи информации поверхность гибкого магнитного диска, имеющего ферромагнитное покрытие, размечается на кольцевые дорожки. Каждая из нескольких десятков дорожек

распределяется на сектора. Информационная емкость сектора составляет несколько сот байт.

При записи информации магнитная головка чтения / записи, подобная магнитофонной, с помощью радиального механизма позиционирования подводится к необходимой дорожке. Когда головка, скользя по поверхности вращающегося диска, оказывается над (или под) соответствующим сектором дорожки, через обмотку сердечника магнитной головки пропускается ток, зависимость величины которого от времени отражает записываемую информацию. Изменения магнитного потока в воздушном зазоре головки приводят к изменениям магнитных свойств покрытия гибкого диска в окрестности кольцевого сектора - носителя записываемой информации.

В режиме чтения записанной информации магнитная головка, скользя по поверхности диска, воспринимает изменения магнитного потока, отражающие накопленную в секторе информацию. В обмотке сердечника головки индуцируется ток, по изменению величины которого во времени можно судить о ранее записанной на диске информации.

ЛИТЕРАТУРА

1. Буховцев В.В., Климонтович Ю.Л., Мякишев Г.Я. Физика. Учебник для 9 класса средней школы. - М.: Просвещение, 1982. - 272 с.
2. Мякишев Г.Я., Буховцев В.В. Физика. Учебник для 10 класса средней школы. - М.: Просвещение, 1983. - 320 с.
3. Левинштейн М.Е., Симин Г.С. Полупроводниковые элементы вычислительной техники. Первое знакомство // Квант, 1985. - # 9. - С. 56-57.
4. Левинштейн М.Е., Симин Г.С. 2. Полевые транзисторы // Квант, 1985. - # 10. - С. 46-47.
5. Левинштейн М.Е., Симин Г.С. 3. Виполярные транзисторы // Квант, 1985. - # 11. - С. 48-49.
6. Левинштейн М.Е., Симин Г.С. 4. Как изготавливают транзисторы в интегральных схемах // Квант, 1985. - # 12. - С. 38-39.
7. Левинштейн М.Е., Симин Г.С. 5. Элементарные логические операции // Квант, 1986. - # 1. - С. 44-45.
8. Левинштейн М.Е., Симин Г.С. 6. Логические схемы на транзисторах // Квант, 1986. - # 2. - С. 44-45.
9. Левинштейн М.Е., Симин Г.С. 7. Элемент памяти - триггер // Квант, 1986. - # 3. - С. 48-49.
10. Левинштейн М.Е., Симин Г.С., Синдаловский В.Я. 8. Регистры // Квант, 1986. - # 4. - С. 46-47.
11. Левинштейн М.Е., Симин Г.С., Синдаловский В.Я. 9. Сумматор // Квант, 1986. - # 5. - С. 44-45.
12. Крутогин Д.Г. Города для электронов // Квант, 1986. - # 2. - С. 6-12. 20.

13. Котов Ю.В. Посмотрим на дисплей // Квант, 1986. - # 4. - С. 43-45.
14. Иткис Г. Кинескопы черно-белого изображения // Радио, 1985. - # 2. - С. 17, цв.вкл.
15. Иткис Г. Кинескопы цветного изображения // Радио, 1985. - # 12. - С. 17, цв.вкл.
16. Иткис Г. Цветной кинескоп с самосведением лучей // Радио, 1988. - # 3. - С. 17, цв.вкл.
17. Сворень Р. Бумажный документ в век безбумажной информатики // Наука и жизнь, 1988. - # 2. - С. 90-96, цв.вкл. 5.
18. Энциклопедический словарь юного техника / Сост. Б.В. Зубков, С.В. Чумаков. - М.: Педагогика, 1980. - 512 с.
19. Энциклопедический словарь юного физика / Сост. В.А. Чужнов. - М.: Педагогика, 1984. - 352 с.
20. Политехнический словарь / Гл. ред. И.И. Артоболовский. - М.: Советская энциклопедия, 1978. - 808 с.
21. Гивоне Д., Россер Р. Микропроцессоры и микрокомпьютеры. - М.: Мир, 1983. - 463 с.
22. Гилмор Ч. Введение в микропроцессорную технику. - М.: Мир, 1984. - 334 с.

Контрольная работа № 1

1. Напишите программу, после исполнения которой в регистре R0 будет содержаться наименьшее из чисел, лежащих в регистрах R0, R1.

2. Напишите программу, после выполнения которой в регистре R0 будет содержаться сумма $500+498+496+\dots+4+2$. Перед началом работы программы R1=500.

3. Какому адресу и какому числу соответствует код I 000 000 000 000 001?

Решения

1. Предполагаем, что программа размещается в памяти, начиная с адреса I500.

сравнить слово R1 с R0	I500
если больше, переход на +1 слово	I502
переслать слово R1 в R0	I504
стоп	I506

2. В регистре R1 будет находиться очередное слагаемое: 500, 498, ..., 4, 2. Когда R1 станет равным нулю, выполнение программы закончится. Располагаем соответствующую программу в памяти ЭВМ, начиная с адреса I500.

очистить слово R0	I500
добавить слово R1 к R0	I502
уменьшить слово R1 на единицу	I504
уменьшить слово R1 на единицу	I506
если больше, переход на -4 слова	I508
стоп	I510

3. Используем соответствие между адресом и кодом:

$$\begin{aligned} \text{адрес} &= i_{15} \cdot 2^{15} + i_{14} \cdot 2^{14} + \dots + i_0 \cdot 2^0 = \\ &= 1 \cdot 2^{15} + 0 \cdot 2^{14} + \dots + 0 \cdot 2^1 + 1 \cdot 2^0 = \\ &= 2^{15} + 2^0 = 32769 \end{aligned}$$

Для получения числа используем соответствие между числом и его кодом:

$$\begin{aligned} \text{число} &= i_{15} \cdot 2^{15} + i_{14} \cdot 2^{14} + \dots + i_0 \cdot 2^0 = i_{15} \cdot 2^{16} \\ &= 1 \cdot 2^{15} + 0 \cdot 2^{14} + \dots + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 2^{16} = \\ &= 2^{15} + 1 = 2^{16} = 32767. \end{aligned}$$

