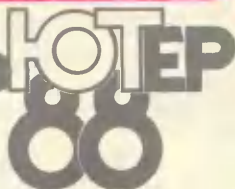


ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР



ДЕТАЛИ О ДЕТАЛЯХ

«Хорошо бы собрать компьютер» — об этом наверняка мечтают многие из наших читателей. Но мечты оставались мечтами — повторить конструкции, которые предлагали журналы «Радио» или «Моделист-конструктор», начинающим радиолюбителям не по плечу — сложные схемы, детали, которых не достать, трудности с налаживанием...

Редакция и решила прийти мечтателям, конечно, деятельным, на помощь. По ее заказу в кружке радиоэлектроники при Раменском Доме пионеров под руководством кандидата технических наук, автора ряда изобретений В. БАРТЕНЕВА, был разработан персональный компьютер, который вы при желании сможете повторить. Дефицитных деталей в нем — минимум. Для удобства вся электронная схема разбита на модули, каждый из которых может работать «сам по себе» в качестве устройств программного управления, мощного программируемого микрокалькулятора или банка данных. В полном варианте микроЭВМ «ЮТ-88» по своим возможностям превосходит большинство популярных среди радиолюбителей компьютеров. А то, что собирать его вы будете постепенно, модуль за модулем, не только упростит сборку, но и позволит освоить программирование, познакомиться с устройством и принципами работы вычислительной техники. Словом, собрав «ЮТ-88», вы научитесь не только им полноценно пользоваться, но и самостоятельно разрабатывать любые собственные конструкции на микропроцессорах. Работа послужит вам хорошей школой в электронике и информатике.

Идеи, которые нашли отражение в нашей публикации, формировались у автора в

1 — процессорный модуль; 2 — дисплейный модуль; 3 — дисплей; 4, 5 — блоки питания процессорного и дисплейного модулей; 6 — клавиатура.



© «ЮТ» для умелых рук», 1989 г.

Персональный компьютер — самоделка серьезная, и не только из-за сложности схемы. Собрать ее как раз не так уж и трудно. Но главная проблема — это где купить детали. Действительно, до сих пор, несмотря на многочисленные выступления журнала «Радио», купить многие микросхемы, особенно серий КР580 и КР541 сегодня проблема. В обычных магазинах радиотоваров они бывают редко. Поэтому тем из нас, кто живет в крупных городах, советуем в первую очередь обратиться в фирменные магазины «Электроника» Министерства электронной промышленности. Многие детали можно найти в магазинах «Юный техник» и «Сделай сам», которые торгуют «чепышками».

А как быть сельским радиолюбителям? В настоящее время в редакцию обратились несколько кооперативов с предложением организовать почтовую торговлю наборами деталей для сборки «ЮТ-88». Так что запаситесь терпением и следите за номерами — скоро мы обязательно опубликуем их адреса. Кроме того, некоторые детали вы можете приобрести через Распределители и Главкоопснабторы. Каталог радиоделателей вы можете взять на почте. Желаем успеха!

ВНИМАНИЕ!

Архитектура, основные схемные решения модулей персонального компьютера «ЮТ-88», программное обеспечение, товарный знак, его тенст и шрифтовое оформление являются исключительной собственностью редакции журнала «Юный техник», представляющей интересы автора. Промышленное и мелкосерийное производство компьютера и его полуфабрикатов, тиражирование программного обеспечения в любой форме без согласия редакции запрещено как государственным, так и кооперативным предприятиям. Но это не исключает сотрудничества.

Приглашаем государственные и кооперативные организации, заинтересовавшиеся персональным компьютером «ЮТ-88», направлять свои предложения на адрес редакции.

процессе обсуждений со специалистами А. Блохиным, С. Аksenовым, И. Грищенко, А. Мелько, за что им благодарны и автор и редакция. Огромное спасибо и кружковцам Г. Бар-

тенева, А. Егорову, Р. Тайцлину, И. Басову, А. Чемелю, А. Дорошкевичу за помощь в макетировании компьютера и отладке программ.

ПЕРВЫЙ ШАГ — МИКРОЭВМ МИНИМАЛЬНОЙ КОНФИГУРАЦИИ



В самом общем виде компьютер, который мы предлагаем вам построить, можно представить в виде структурной схемы, изображенной на рисунке 1. В ней приведены три основных элемента — центральный процессор, память и устройство ввода/вывода. Эти три блока входят в состав любого компьютера.

Центральный процессор — самый сложный элемент. Он представляет собой программно-управляемое устройство, обрабатывающее информацию и осуществляющее обмен ею между основными компонентами ЭВМ. В качестве центрального процессора в нашем компьютере используем большую интегральную схему (БИС) микропроцессора КР580ВМ80А. Выбор наш не случаен. Этот микропроцессор, думаем, еще долгое время будет популярным среди радиолюбителей из-за большого объема программного обеспечения, разработанного под его систему команд. А наличие у него фиксированной и простой системы команд облегчает составление программ даже в машинных кодах. И, наконец, что очень важно, этот микропроцессор уже появился в продаже.

Память микроЭВМ состоит из постоянного запоминающего устройства (ПЗУ), допускающего только считывание хранимой информации, и оперативного запоминающего устройства (ОЗУ), работающего как в режиме чтения информации из ОЗУ, так и записи в него новых данных. Фактически память — это совокупность однотипных ячеек, в каждой из которых хранится закодированная информация. Единичной измерения ее объема является бит. Для хранения 1 бита достаточно иметь одноразрядную ячейку, в которую может быть записан 0 или 1. Разрядность ячейки памяти микроЭВМ определяется разрядностью шины данных микропроцессора (для микропроцессора КР580ВМ80А составляет 8 бит, или 1 байт).

Каждая ячейка памяти имеет свой номер — адрес. Число непосредственно адресуемых ячеек определяется разрядностью адреса, формируемого микропроцессором. В нашем случае это 16 разрядов, что позволяет иметь максимальный объем адресуемой памяти чуть больше 65 тысяч ячеек.

Устройства ввода/вывода включают в себя клавиатуру, дисплей и другие устройства, например, принтер. Информация между ними и микропроцессором передается через так называемые порты ввода/вывода. К микропроцессору КР580ВМ80А можно подключить в общей сложности 256 таких устройств.

Объем используемой памяти, число устройств ввода/вывода определяют конфигурацию компьютера, его возможности и стоимость. Так что мы вам предлагаем собрать компьютер, который имеет изменяемую конфигурацию. В своем минимальном исполнении, когда используется только процессорный модуль — это простая в изготовлении, дешевая однопланная микроЭВМ на доступных микросхемах. Однако несмотря на мини-

мальный объем ОЗУ и ПЗУ, простейшие клавиатуру и дисплей, наш первый модуль поможет вам не только разобраться в работе микроЭВМ, но и научиться писать программы. Заметим, что это лишь наш первый этап в освоении компьютерной техники. Изготовив модуль и освоив его программирование, мы будем готовы сделать следующий шаг — расширить конфигурацию нашей микроЭВМ, добавив дисплейный модуль с полной клавиатурой и квазидиск произвольного объема.

Так, шаг за шагом, от простого к сложному, мы с вами пройдем путь от простейшей микроЭВМ к настоящему персональному компьютеру с мощным программным обеспечением на основе операционной системы CP/M.

БЛОК-СХЕМА МИКРОЭВМ

Давайте более подробно рассмотрим структурную схему на рисунке 1. Раскроем ее составные части.

Шины — это совокупность проводов, объединенных между собой по функциональному признаку. Восемь проводов образуют шину данных, шестнадцать — шину адреса, пять — шину управления. Такая архитектура позволяет легко наращивать различные периферийные устройства, присоединяя их к персональной ЭВМ.

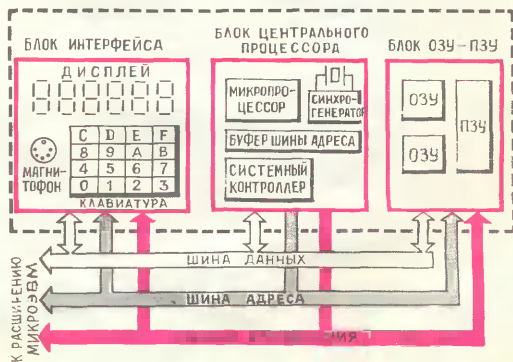
Блок-схема первого модуля условно разбита на три блока: блок центрального процессора, блок памяти и блок интерфейса.

Такая разбивка его схемы на блоки сделана, исходя из удобства объяснения принципа работы отдельных узлов компьютера. Но, учитывая функциональную законченность каждого блока, вы можете использовать их по своему усмотрению в различных радиодобительских конструкциях.

В состав блока центрального процессора на рисунке 1 входят микропроцессор и дополнительные элементы, обеспечивающие его работу, — генератор тактовых импульсов, формирователь сигналов шины данных и шины управления (системный контроллер), буфер шины адреса. Блок памяти, как сказано, состоит из ПЗУ и ОЗУ. Блок интерфейса — простейшая клавиатура с семнадцатью кнопками, дисплей на семисегментных индикаторах и схема сопряжения с кассетным магнитофоном.

В компьютере передаваемые сигналы представлены двумя уровнями напряжения — логическими «0» и «1». Сигнал, который вызывает выполнение некоторого действия, называют активным. Активное состояние может быть при логической единице либо при логическом нуле. В частности, на шине управления сигналы активны при уровне логического «0». Таких управляющих сигналов пять: ЧТЗУ — чтение памяти; ЗПЗУ — запись в память; ЧТВВ — чтение порта ввода/вывода; ЗПВВ — запись в порт ввода/вывода; ППР — подтверждение прерывания. Шина

Рисунок 1.



управления используется лишь для вывода сигналов управления, то есть она однонаправленная. Основная причина введения активных сигналов низкого уровня (они помечены чертой сверху) состоит в простоте объединения их по ИЛИ. Кроме того, выходной каскад с таким сигналом большую часть времени находится в выключенном состоянии и потребляет меньше энергии от источника питания.

Как и шина управления, шина адреса лишь передает выходные сигналы микропроцессора. Отдельные линии шины адреса имеют обозначение от A0 до A15.

В отличие от шин управления и адреса, шина данных — двунаправленная. Передача данных здесь может производиться как от процессора, так и к процессору. Однако в каждый данный момент времени она осуществляется только в одном направлении. Отдельные линии шины данных обозначаются от D0 до D7.

По шинам компьютера информация передается в двоичном коде. Но его запись кодов требует много места и не всегда удобна. В дальнейшем мы будем пользоваться шестнадцатиричной формой представления кодируемых сигналов, то есть системой счисления с основанием 16. В ней используются символы: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E и F. В таблице 1 приведены примеры представления чисел в двоичной, шестнадцатиричной и десятичной системах счисления.

Таблица 1.

Шестнадцатиричное число	Двоичное число	Десятичное число
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15
10	10000	16
11	10001	17

Процедура преобразования двоичного числа в шестнадцатиричное проста. Разряды, начиная с младшего, разбиваются на группы по четыре. Каждой группе подбирается соответствующий шестнадцатиричный символ. В частности, диапазон адресов компьютера составляет 0000₁₆ — FFFF₁₆ (в дальнейшем основание системы счисления мы указывать не будем, а в конце каждого кода будем добавлять букву H, например: адресное пространство компьютера составляет 0000H — FFFFH). Шестнадцатиричная форма записи очень проста, да и воспринимать ее легче, чем двоичную. На дисплее, после соответствующего преобразования в дешифраторе двоичного кода в шестнадцатиричный, будут индироваться шестнадцатиричные цифры. Ввод же с клавиатуры производится в шестнадцатиричных цифрах, хотя затем код каждой нажатой шестнадцатиричной клавиши будет преобразован дешифратором в двоичный код, подаваемый на шину данных.

На рисунке 2 отсутствует один блок, о котором нельзя не упомянуть, — это блок формирования выборки кодов адреса. Все внешние устройства микроЭВМ могут быть подключены к микропроцессору только при выполнении двух условий: на них должен быть подан управляющий сигнал с шины управления и это устройство должно быть выбрано соответствующим адресом. Формирование сигналов выборки кода адреса (BK) внешних устройств осуществляется дешифрацией сигналов шины адреса.

В нашем первом модуле применяются два способа ввода/вывода информации. Изолированный — для портов клавиатуры (A0H) и схемы сопряжения с касетным магнитофоном (A1H). Отображенный на память для портов дисплея с адресами: 9000H (порт младшего байта дисплея), 9001H, 9002H (порты старших байтов дисплея). Как уже отмечалось, число изолированных портов ввода/вывода не превышает 256, и их адреса изолированы от адресного пространства памяти. В микропроцессоре KP580 имеется две команды ввода: IN и вывода OUI, при выполнении которых данные из адресного порта загружаются в микропроцессор (ввод) или, наоборот, из микропроцессора передаются в адресный порт (вывод). Ввод/вывод, отображенный на память, характеризуется тем, что процессор обращается к портам как к ячейкам адресного пространства памяти и все команды, содержащие адреса портов, преобразуются в команды ввода/вывода. В нашем микроЭВМ используются оба эти способа

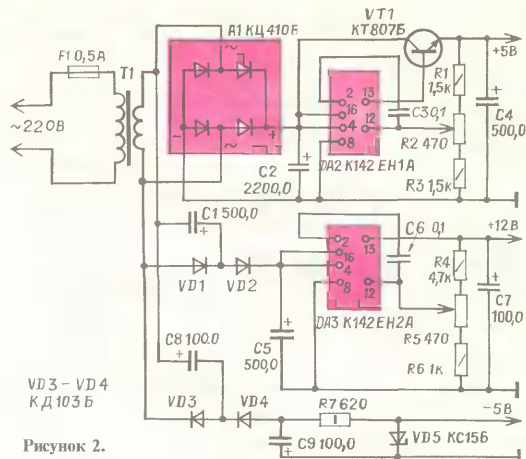


Рисунок 2.

ИСТОЧНИК ПИТАНИЯ

На нашей блок-схеме для простоты картины отсутствует связь модуля с источниками питания, тем не менее нужно сказать и о них. Для питания нашей микроЭВМ необходимы три источника постоянного напряжения: +5B, +12B и -5B. Пятивольтовый источник, наиболее мощный, должен обеспечивать ток до 1 А. Ток потребления по цепям +12B не превышает 50 мА, а по цепям -5B — 5 мА. Напряжение +12B, -5B используется для питания микропроцессора. Сборку микроЭВМ необходимо начинать с изготовления модуля питания. Его электрическая схема показана на рис. 2.

Переменное напряжение (порядка 8—10 В) вторичной обмотки поступает на мостовой выпрямитель A1 и на два удвоителя напряжения на диодах VD1 — VD4. Выпрямленное напряжение сглаживается емкостью C2 и стабилизируется с помощью микросхемы DA2.

Увеличение нагрузочной способности стабилизатора достигается подключением внешнего транзистора VT1, установленного на радиаторе. С помощью потенциометра R2 устанавливается требуемое выходное напряжение. Выпрямленные и удвоенные напряжения положительной и отрицательной полярности сглаживаются соответственно емкостями C5 и C9. Положительное напряжение стабилизируется с помощью микросхемы DA3. Значение выходного напряжения +12B регулируется с помощью потенциометра R5. Отрицательное напряжение -5B снимается с параметрического стабилизатора на стабилитроне VD5.

В качестве трансформатора T1 может быть использован любой понижающий трансформатор со вторичной обмоткой, обеспечивающей напряжение порядка 8—10 В и ток до 2 А. Его можно изготовить и самостоятельно, намотав на сердечнике ШП16х32. Сетевая обмотка содержит 1350 витков провода ПЭВ-2 диаметром 0,31 мм, вторичная обмотка содержит 50 витков того же провода диаметром 1,5 мм.

ЭЛЕКТРИЧЕСКАЯ СХЕМА

Для удобства мы решили поместить электрическую схему первого модуля компьютера «ЮТ-88» на вкладке.

Блок центрального процессора выполнен на микросхемах DD1, DD2, DD3, DD6 и DD8. Микросхема DD1 — микропроцессор KP580BM80A. Микросхема DD2 KP580G F24 предназначена для формирования двух последовательных тактовых импульсов с требуемыми временными и амплитудными параметрами.

Она содержит задающий генератор, стабилизированный внешним кварцевым резонатором ZQ1 с частотой 16 МГц. Кроме формирователя синхроимпульсов Ф1 и Ф2, микросхема DD2 содержит триггеры синхронизации сигналов начальной установки и готовности, а также схему формирования строба состояния STB, используемого для записи бита состояния при формировании сигналов шины управления.

Адресная шина (ША) подключается ко многим устройствам.

Выходы же микропроцессора KP580BM80A выдерживают лишь нагрузку одного входа TTL микросхем. Поэтому для повышения нагрузочной способности к шине адреса подключены буферные регистры DD5 и DD6 KP580IP82.

Микросхема KP580IP82 представляет собой 8-разрядный регистр с управляемыми выходными формирователями. Запись информации происходит при наличии логической единицы на выводе 11 STB, а выдана информация — при подаче на вывод 9 OE уровня логического нуля.

Увеличение нагрузочной способности двунаправленной шины данных выполнено с помощью микросхемы системного контроллера DD8 KP580BK38. Она осуществляет формирование выходных сигналов не только шины данных, но и шины управления. Микросхема содержит двунаправленный магистральный формирователь, регистр состояния и схему формирования управляющих сигналов. Следующий блок микроЭВМ предназначен для кратковременного и долговременного хранения данных и программ. На схеме модуля «ОТ-88» блок памяти представлен микросхемами ПЗУ KP556PT5 (DD10, DD11) и микросхемами ОЗУ KP541PY2 (DD13, DD14).

Микросхемы DD10, DD11 предназначены для хранения управляющей программы МОНИТОР. Подробно о содержимом ПЗУ будет рассказано в разделе, посвященном программному обеспечению микроЭВМ. Сейчас же рассмотрим, как происходит считывание информации с микросхем ПЗУ.

После нажатия на кнопку «УСТ» на шине адреса устанавливается адрес 0000H, а на шине управления — сигнал ЧТЗУ, который после инвертирования элементом DD21.5 подается на вход V3 микросхем ПЗУ. Дешифратор на микросхеме K155ИД3 (DD7) вырабатывает сигналы выборки адреса, разбивая всю доступную область адресного пространства микроЭВМ на сегменты по 4 Кбайта каждый. При этом сигнал выборки сегмента адресного пространства 0000H—0FFFH действует на выводе 1 микросхемы DD7 и после инвертирования элементом DD9.1 подается на входы V4 микросхем ПЗУ DD10 и DD11. Учитывая, что на два других входа V1, V2 микросхем ПЗУ DD10 и DD11 подаются сигналы с адресной шины A11 и A9, информация с микросхемы DD10 будет считываться в области адресного пространства 0000H—0FFFH. В отличие от DD10 на вход выборки V2 микросхемы DD11 сигнал с адресной шины A9 подается после инвертирования элементом DD9.2, поэтому в области адресного пространства 0200H—03FFFH информация будет считываться с микросхемы DD11.

Сигналы выборки сегментов адресного пространства 0000H—0FFFH, E000H—EFFFH и F000H—FFFFH вы используете при расширении конфигурации компьютера. Поэтому вместе с сигналами управления они выведены на внешний разъем.

Схема ОЗУ выполнена на микросхемах DD13, DD14 типа KP541PY2, имеющих совмещенные вход и вывод. Объем ОЗУ 1 Кбайт. При записи данных в ОЗУ общие выходы 11—14 микросхем действуют как входы. Чтобы избежать конфликта на внутренних выходных линиях памяти при вводе данных в ОЗУ, когда сигнал управления ПЗУ имеет уровень логического 0, внутренние выходы ОЗУ блокируются с помощью внутренних схем. Указанные внутренние выходы переводятся при этом в состояние высокого сопротивления. Во время выполнения операции записи данных в ОЗУ в состоянии логического 0 должны находиться как выходы WE разрешения записи, так и выходы 8 CS выбора микросхемы. При чтении данных из ОЗУ выходы 8 CS выбора микросхемы должны иметь состояние логического 0, а выходы 10 WE разрешения записи — состояние логической 1. Соответствующая схема, предназначенная для формирования требуемых управляющих сигналов, выполнена на DD9.4, DD12.1 и DD12.2. ОЗУ работает в адресном пространстве C000H—C3FFFH.

В состав нашего первого модуля, помимо процессорного блока и блока памяти, входит блок интерфейса, с помощью которого осуществляется ввод и вывод данных. Ввод производится с клавиатуры или с кассетного магнитофона. Вывод — путем индикации на дисплее или записью их на кассетный магнитофон для долговременного хранения.

Содержимое ячеек памяти с адресами 9000H, 9001H и 9002H отображается на индикаторах в виде шестнадцатичных цифр.

Программная реализация интерфейса с кассетным магнитофоном предельно упростила его сопряжение с микроЭВМ и обеспечила программную совместимость по вводу данных с кассетного магнитофона с микроЭВМ «Микро-80» и «PK-86».

Рассмотрим электрическую схему модуля в части блока интерфейса. В индикаторе ячейки памяти выполнены на регистрах DD18 и DD19 типа K155PН. Каждая микросхема представляет собой восемь четырехразрядных ячеек, адресуемых независимо как при записи, так и при считывании из них данных. Входы дешифрации регистров при записи WA, WB подключены к адресам AO и A1 шины адреса. Запись данных осуществляется подачей логического 0 на вход C1. Этот сигнал образуется с помощью элементов DD9.3, DD9.5 и DD12.3 объединением кода выборки адреса, соответствующего области адресов 9000H—9FFFH, дешифратора DD7 и управляющего сигнала ПЗУ. Входы дешифраторов чтения регистров DD18 и DD19 VA и VB под-

ключены к старшим разрядам двоичного счетчика на микросхеме DD24 типа K155ИЕ5. Чтение данных производится при подаче на вход C2 логического нуля. Выходы 4-х разрядных регистров DD18 и DD19 выполнены на транзисторах с открытым коллектором и объединяются между собой.

В схему управления динамической индикацией входит генератор синхронизмпульсов на элементах DD21.3, DD21.4, двоичный счетчик DD24 и дешифратор управляющих импульсов на микросхеме K155ИД4 (DD20). Двоичный код преобразуется в дешифраторе на микросхеме DD23 (K155PE3) в специальный код сегментного индикатора. Выводы DD23, идущие к сегментам, подключены через резисторы сопротивлением 1 кОм к шине питания +5 В. При этом на соответствующем разряде индикатора с общим катодом (например, А, LC324А или АLC318) должен быть уровень логического нуля. Содержимое дешифратора (DD23) двоичного кода в семисегментный шестнадцатичный приведено в таблице 2.

Таблица 2. СОДЕРЖИМОЕ ДЕШИФРАТОРА K155PE3 (DD23)

Цифра индикатора	Адрес K155PE3 код на входе	Код ПЗУ
0	00	3F
1	01	06
2	02	5B
3	03	4F
4	04	66
5	05	6D
6	06	7D
7	07	07
8	08	7F
9	09	6F
A	0A	77
B	0B	7C
C	0C	39
D	0D	5E
E	0E	79
F	0F	71

Клавиатура состоит из 17 кнопок, соединенных с дешифратором, на выходе которого формируется двоичный код нажатой клавиши. Дешифратор кода нажатой клавиши выполнен на микросхемах ПЗУ DD15 и DD16 типа KP556PT4. Содержимое этих микросхем приведено в таблице 3. Когда не нажата ни одна из клавиш, при поступлении сигнала выборки клавиатуры в виде логического нуля, обе микросхемы ПЗУ открываются, и поскольку по адресу FF в микросхемах ПЗУ записаны нули, то они и считаются на шину данных.

Таблица 3. СОДЕРЖИМОЕ ПЗУ КЛАВИАТУРЫ

Адрес на входе	Содержимое ПЗУ (DD15)	Содержимое ПЗУ (DD16)
FE	0	7
FF	0	8
FD	1	9
FB	2	A
F7	3	B
EF	4	C
DF	5	D
BF	6	E
7F	7	F
Все остальные адреса	0	0

Код нулевой клавиши 10H формируется вспомогательным логическим элементом 2И—НЕ с открытым коллектором микросхемы DD17. Аналогично формируется код клавиши «ШАГ НАЗАД». Коды остальных 15 клавиш формируются микросхемами DD15 и DD16. Причем при формировании кодов от «до» («») логическая единица в разряде D3 шины данных запрещает выборку микросхемы DD15. Две дополнительные клавиши могут быть задействованы при подключении их к выводам 4 микросхемы DD17 (код 20H) или 10 микросхемы DD17 (код 40H).

Для сопряжения с кассетным магнитофоном используется D-триггер DD4.2, информационный вход которого подключен к нулевому разряду шины данных, а выход через фильтр соединяется со входом кассетного магнитофона в режиме записи. В режиме воспроизведения выход кассетного магнитофона подключается ко входу операционного усилителя DA1. Выход операционного усилителя через схему формирования импульсов положительной полярности подключается к одному входу логического элемента DD12.4 с открытым коллектором. Второй вход этого логического элемента подключен к схеме объединения по ИЛИ на микросхеме DD22 управляющего сигнала ЧТВВ и сигналов выборки кода порта A1H с помощью дешифратора DD7 и инверсного адреса A0. Выход элемента DD12.4 подключен к нулевому разряду шины данных.

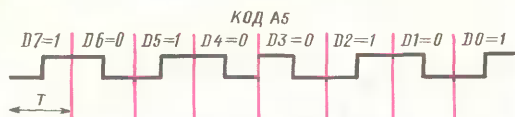


Рисунок 3.

Запись и чтение информации при работе с кассетным магнитофоном производится последовательным кодом, бит за битом.

В основу записи информации на магнитную ленту положен метод двухфазного кодирования. Формирование последовательности двухфазных кодов производится программно. На рисунке 3 показано, как выглядит последовательный двухфазный код после преобразования байта А5 (его двоичное представление 10100101). Запись байта при последовательном двухфазном кодировании начинается со старшего разряда.

В середине передаваемого бита происходит изменение его значения на противоположное, причем изменение с «1» на «0» означает, что передан бит, равный 0, а обратное изменение с «0» на «1» — бит, равный единице.

На границе двух одинаковых по значению смежных битов всегда происходит изменение значения двухфазного кода. На границе разных по значению смежных битов изменение двухфазного кода не происходит. Период следования информации T выбирается из соображений надежности считывания информации. Если $T=0,66$ мс, то скорость записи считывания равна 1500 бит/сек, что позволяет записать на одну сторону стандартной кассеты типа МК-60-2 до 256 Кбайт информации. Таким образом интерфейс с кассетным магнитофоном представляет собой односторонний порт ввода-вывода.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Ограниченный объем ПЗУ и ОЗУ в микроЭВМ минимальной конфигурации предъявляет особые требования к эффективности программного обеспечения, которое фактически ограничено программой МОНИТОР. Однако, даже несмотря на небольшой объем, МОНИТОР «ЮТ-88» даст вам возможность писать и отлаживать программы в машинных кодах, модифицировать их за счет вставки или удаления отдельных команд, пересылать отлаженные программы в новые адреса и корректировать их. При программировании в машинных кодах удается максимально приблизиться к аппаратным средствам микроЭВМ, учесть особенности микропроцессора и каждой отдельной команды и в результате получить предельно короткие и быстродействующие программы. Для программирования в машинных кодах требуются только карандаш и бумага. Составленную программу можно сразу с помощью МОНИТОРА загрузить в ОЗУ микроЭВМ и попробовать в работе. Используя при написании таких программ символическое обозначение команд в виде мнемоник языка Ассемблера с последующим ручным ассемблированием с помощью компактных табличных представлений команд/мнемоник-команд/кодов, удается по существу освоить программирование на языке Ассемблера.

Прежде чем перейти к рассмотрению директив МОНИТОРА, целесообразно познакомиться с системой команд и устройством микропроцессора КР5801К80А в справочном листке «ЮТ-88».

Все команды микроЭВМ сведены в таблицу (см. вкладку). Выделим наиболее общие закономерности их использования.

1. Арифметические и логические операции разрешены только между аккумулятором и байтом данных или между аккумулятором и любым регистром.

2. Аккумулятор и регистровая пара HL являются единственными регистрами, которые могут быть непосредственно загружены в память.

3. Аккумулятор является единственным регистром, который может быть инвертирован, сдвинут, косвенно загружен в память с использованием регистровых пар BC и DE или использован в командах ввода/вывода (IN, OUT).

4. Регистровая пара HL является единственной, содержимое которой может быть передано в счетчик команд (команда PCNL) или указатель стека (команда SPHL).

5. Регистровая пара H может использоваться как аккумулятор двойной длины при сложении 16 разрядных чисел (команда DAD).

6. Содержимое регистровых пар HL и DE можно менять местами (команда HCHG).

7. Отдельные команды могут применяться для специальных функций. Команды XRA A или SUB A обнуляют аккумулятор. Команды ANA A или ORA A очищают флаг переноса, а команда ADD A выполняет логический сдвиг аккумулятора влево.

8. Команды увеличения (уменьшения) на 1 регистров INR (DCR) действуют на все флаги, за исключением флага переноса. Команды увеличения (уменьшения) на 1 регистровых пар INX (DCX) не оказывают влияния на флаги.

9. В стек или из стека могут быть переданы только регистровые пары. Одной из таких регистровых пар является слово состояния микропро-

цессора, которое содержит аккумулятор (старший байт) и флаги (младший байт). Команды CALL и RET передают адрес в стек и обратно.

10. При записи 16-разрядных адресов младший байт записывается первым.

Целесообразно также отдельно рассмотреть основные способы адресации данных в микроЭВМ. Предусмотрено 3 способа адресации: прямая, непосредственная и косвенная.

1. Прямая адресация (из памяти или в память с конкретным адресом) может использоваться для работы с аккумулятором и регистрами H—L. Адрес данных записывается во втором и третьем байте команды. Например:

LDA C000H

Эта команда загружает А из ячейки памяти с адресом C000H;

STA 9000H

Эта команда загружает ячейку памяти с адресом 9000H содержимым аккумулятора.

LHLD C000H

Эта команда загружает регистр L из ячейки с адресом C000H, а регистр H из ячейки с адресом C001H, то есть сначала младший байт, а затем старший байт.

SHLD 9001H

Эта команда загружает в ячейку памяти 9001H содержимое регистра L, а в ячейку памяти 9002H регистра H.

2. Непосредственная адресация, то есть конкретное значение, записывается в байте, следующем за командой.

Например, MVI A, 11H.

Эта команда загружает в А значение 11H;

LXI H, 9001H

Эта команда загружает в регистры H—L значение 9001H соответственно.

3. Косвенная адресация (по адресу в регистровой паре HL, DE или BC). Например, MOV A, M или MOV M, A.

Команды загрузки регистра из ячейки памяти с адресом в регистровой паре HL или наборот.

LDAX D или LDAX B

Команды загрузки аккумулятора из памяти по адресам в регистровых парах DE или BC.

STAX D или STAX B

Команды загрузки памяти по адресу в регистровых парах DE или BC содержимым аккумулятора.

Используя команды с разными способами адресации, приведем программу вывода единиц на все индикаторы:

1. 2-байтная команда	MVI A, 11H	Занесение константы в аккумулятор
2. 3-байтная команда	STA 9000H	Вывод «1» на правый индикатор
3. 3-байтная команда	LXI H, 9001H	Занесение адреса среднего индикатора в H
4. Однобайтная команда	MOV M, A	Вывод на средний индикатор «1»
5. 3-байтная команда	LXI D, 9002H	Занесение адреса левого индикатора в DE
6. Однобайтная команда	STAX D	Вывод «1» на левом индикаторе

Программа содержит шесть команд, и, чтобы ее записать в память, потребуется 13 ячеек памяти. Используя таблицу, находя в ней мнемонику каждой команды, определим соответствующие им коды.

1. MVI A,	11H	3E
2. STA	9000H	32
		00
		90
3. LXI H,	9001H	21
		01
		90
4. MOV M,A		77
5. LXI D,	9002H	11
		02
		90
6. STAX D		12

Если эти коды записать в ОЗУ микроЭВМ, то в результате исполнения этой программы на всех индикаторах микроЭВМ выветятся единички.

МОНИТОР — УПРАВЛЯЮЩАЯ ПРОГРАММА МИКРОЭВМ

Основой программного обеспечения микроЭВМ, как бы его ядром, является управляющая программа МОНИТОР объемом 1 Кбайт. Несмотря на столь малый объем, в ней имеются директивы записи, чтения ОЗУ, пуска программ с задаваемого адреса, тестирования ОЗУ и индикаторов, записи и чтения программ с кассетного магнитофона с их верификацией. Имеется возможность пошаговой коррекции содержимого ОЗУ, осуществления перемещение программ в новые адреса с коррекцией адресов программы, вычисления контрольной суммы, вставления и удаления байты команд с коррекцией адресов остальной части программы. Минимизация объема МОНИТОРА достигнута прежде

В приводимой таблице команд микропроцессора KP5801K80 исполняются следующие обозначения:
 [] регистр или регистровая пара
 [] ячейка памяти, адресная регистровой парой
 → направления передачи данных
 DATA [8] байт данных
 DATA [16] два байта данных
 ADDR [16] 16-ти разрядный адрес ячейки памяти

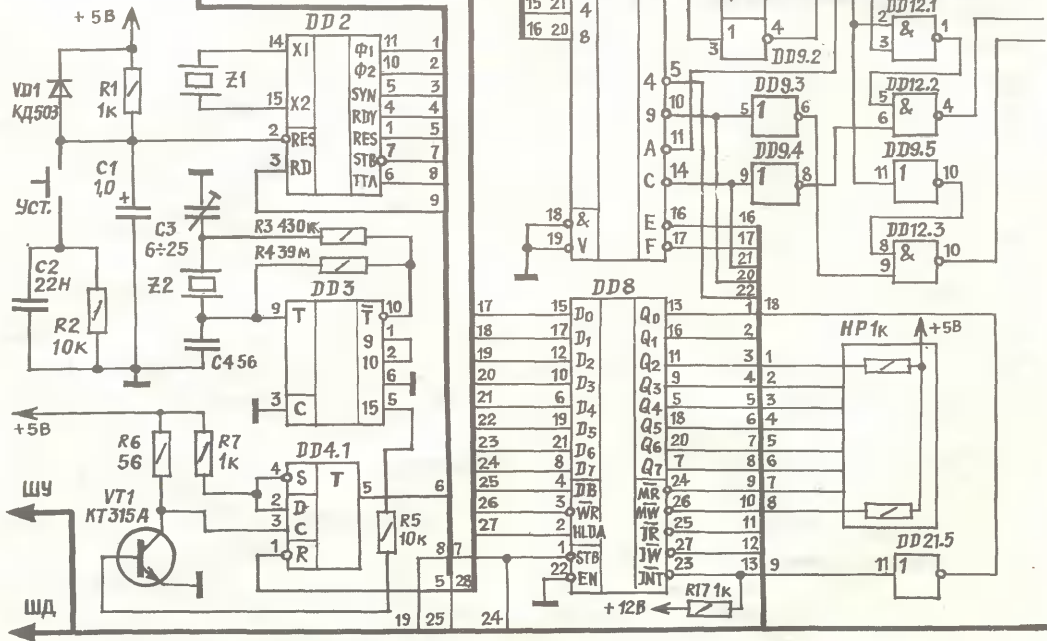
0
 00
 FLAG *
 *
 PCL младший байт содержимого PC
 PCN старший байт содержимого PC
 BU внешнее устройство

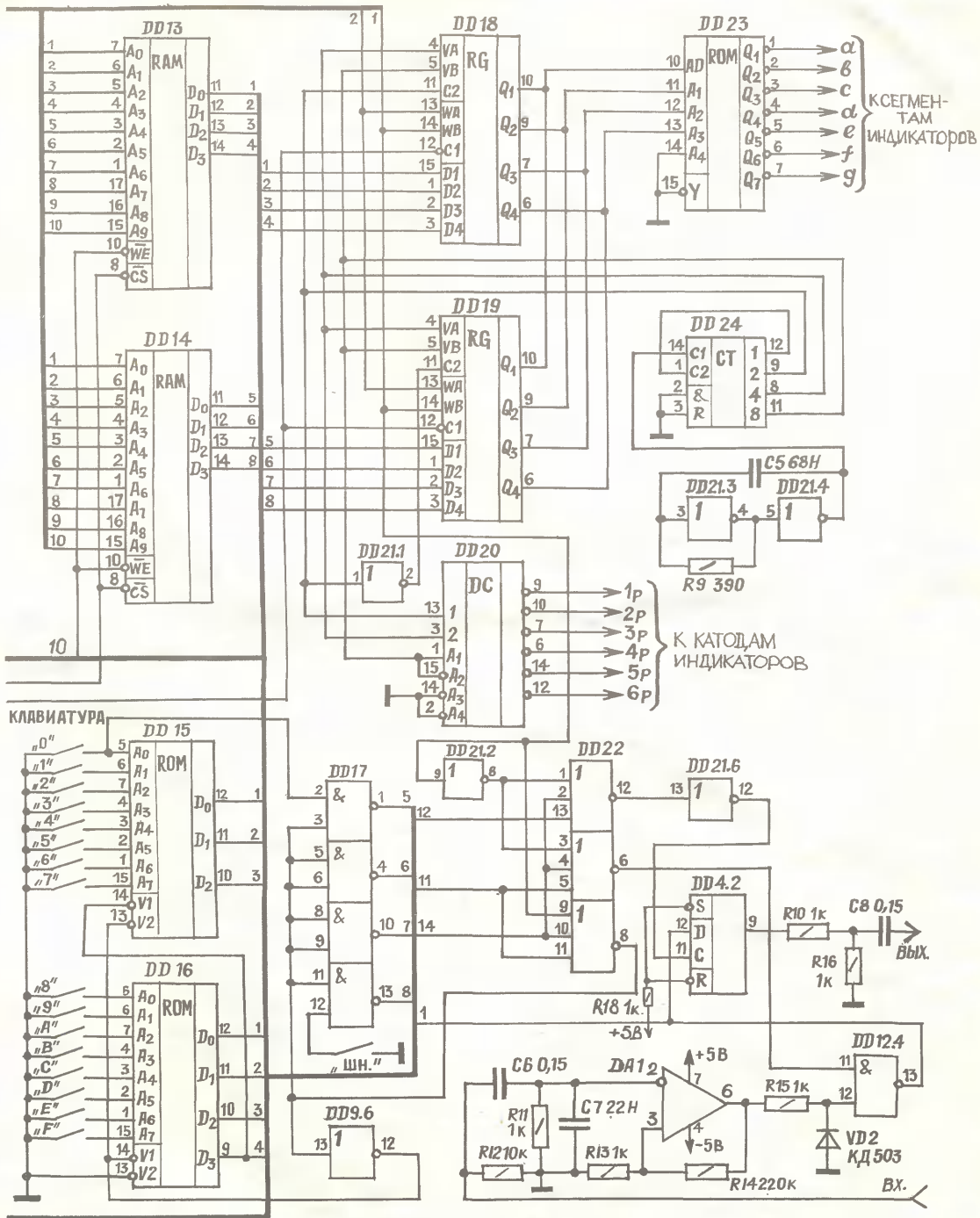
Код команды	Мнемоника команды	Действие команды	Содержимое регистра признаков S'Z'AC'P'C	Число тактов
1	2	3	4	5
Команды пересылки данных				
7F	MOV A,A	[A]←[A]	---	5
78	MOV A,B	[A]←[B]	---	5
79	MOV A,C	[A]←[C]	---	5
7A	MOV A,D	[A]←[D]	---	5
7B	MOV A,E	[A]←[E]	---	5
7C	MOV A,H	[A]←[H]	---	5
7D	MOV A,L	[A]←[L]	---	5
47	MOV B,A	[B]←[A]	---	5
40	MOV B,B	[B]←[B]	---	5
41	MOV B,C	[B]←[C]	---	5
42	MOV B,D	[B]←[D]	---	5
43	MOV B,E	[B]←[E]	---	5
44	MOV B,H	[B]←[H]	---	5
45	MOV B,L	[B]←[L]	---	5
4F	MOV C,A	[C]←[A]	---	5
48	MOV C,B	[C]←[B]	---	5
49	MOV C,C	[C]←[C]	---	5
4A	MOV C,D	[C]←[D]	---	5
4B	MOV C,E	[C]←[E]	---	5
4C	MOV C,H	[C]←[H]	---	5
4D	MOV C,L	[C]←[L]	---	5
57	MOV D,A	[D]←[A]	---	5
58	MOV D,B	[D]←[B]	---	5
59	MOV D,C	[D]←[C]	---	5
5A	MOV D,D	[D]←[D]	---	5
5B	MOV D,E	[D]←[E]	---	5
5C	MOV D,H	[D]←[H]	---	5
5D	MOV D,L	[D]←[L]	---	5
5F	MOV E,A	[E]←[A]	---	5
58	MOV E,B	[E]←[B]	---	5
59	MOV E,C	[E]←[C]	---	5
5A	MOV E,D	[E]←[D]	---	5
5B	MOV E,E	[E]←[E]	---	5
5C	MOV E,H	[E]←[H]	---	5
5D	MOV E,L	[E]←[L]	---	5
67	MOV H,A	[H]←[A]	---	5
60	MOV H,B	[H]←[B]	---	5
61	MOV H,C	[H]←[C]	---	5
62	MOV H,D	[H]←[D]	---	5
63	MOV H,E	[H]←[E]	---	5
64	MOV H,H	[H]←[H]	---	5
65	MOV H,L	[H]←[L]	---	5
6F	MOV L,A	[L]←[A]	---	5
68	MOV L,B	[L]←[B]	---	5
69	MOV L,C	[L]←[C]	---	5
6A	MOV L,D	[L]←[D]	---	5
6B	MOV L,E	[L]←[E]	---	5
6C	MOV L,H	[L]←[H]	---	5
6D	MOV L,L	[L]←[L]	---	5
7E	MOV A,M	[A]←[[HL]]	---	7
46	MOV B,M	[B]←[[HL]]	---	7
4E	MOV C,M	[C]←[[HL]]	---	7
56	MOV D,M	[D]←[[HL]]	---	7
5E	MOV E,M	[E]←[[HL]]	---	7
66	MOV H,M	[H]←[[HL]]	---	7
6E	MOV L,M	[L]←[[HL]]	---	7
77	MOV M,A	[[HL]]←[A]	---	7
70	MOV M,B	[[HL]]←[B]	---	7
71	MOV M,C	[[HL]]←[C]	---	7
72	MOV M,D	[[HL]]←[D]	---	7
73	MOV M,E	[[HL]]←[E]	---	7
74	MOV M,H	[[HL]]←[H]	---	7
75	MOV M,L	[[HL]]←[L]	---	7
3E	MVI A,DATA[8]	[A]←DATA[8]	---	7
06	MVI B,DATA[8]	[B]←DATA[8]	---	7
0E	MVI C,DATA[8]	[C]←DATA[8]	---	7
16	MVI D,DATA[8]	[D]←DATA[8]	---	7
1E	MVI E,DATA[8]	[E]←DATA[8]	---	7
26	MVI H,DATA[8]	[H]←DATA[8]	---	7
2E	MVI L,DATA[8]	[L]←DATA[8]	---	7
36	MVI H,DATA[8]	[[HL]]←DATA[8]	---	10
01	LXI B,DATA[16]	[[BC]]←DATA[16]	---	10
11	LXI D,DATA[16]	[[DE]]←DATA[16]	---	10
21	LXI H,DATA[16]	[[HL]]←DATA[16]	---	10
31	LXI SP,DATA[16]	[[SP]]←DATA[16]	---	10
3A	LDA ADDR[16]	[A]←ADDR[16]	---	13
32	STA ADDR[16]	ADDR[16]←[A]	---	13

1	2	3	4	5
2A	LHLD ADDR[16]	[L]←ADDR[16] [H]←ADDR[16]+1	---	16
22	SHLD ADDR[16]	ADDR[16]←[L] ADDR[16]+1←[H]	---	16
0A	LDAX B	[A]←[[BC]]	---	7
1A	LDAX D	[A]←[[DE]]	---	7
02	STAX B	[[BC]]←[A]	---	7
12	STAX D	[[DE]]←[A]	---	7
E8	XCHG	[HL]←[DE] [SP]←[HL]	---	4
F9	SPHL	[L]←[[SP]]	---	5
E3	XTHL	[H]←[[SP]+1] [SP]←[[SP]-1]	---	18
C5	PUSH B	[[SP]-1]←[B] [[SP]-2]←[C] [SP]←[[SP]-2]	---	11
D5	PUSH D	[[SP]-1]←[D] [[SP]-2]←[E] [SP]←[[SP]-2]	---	11
E5	PUSH H	[[SP]-1]←[H] [[SP]-2]←[L] [SP]←[[SP]-2]	---	11
F5	PUSH PSW	[[SP]-1]←[A] [[SP]-2]←[FLAG] [SP]←[[SP]-2]	---	11
C1	POP B	[C]←[[SP]] [B]←[[SP]+1] [SP]←[[SP]+2]	---	11
D1	POP D	[E]←[[SP]] [D]←[[SP]+1] [SP]←[[SP]+2]	---	11
E1	POP H	[L]←[[SP]] [H]←[[SP]+1] [SP]←[[SP]+2]	---	11
F1	POP PSW	[FLAG]←[[SP]] [A]←[[SP]+1] [SP]←[[SP]+2]	---	11
DB	IN адрес ВУ	[A]←[ВУ]	---	10
D3	OUT адрес ВУ	[ВУ]←[A]	---	10
Арифметико-логические команды				
87	ADD A	[A]←[A]+[A]	***	4
80	ADD B	[A]←[A]+[B]	***	4
81	ADD C	[A]←[A]+[C]	***	4
82	ADD D	[A]←[A]+[D]	***	4
83	ADD E	[A]←[A]+[E]	***	4
84	ADD H	[A]←[A]+[H]	***	4
85	ADD L	[A]←[A]+[L]	***	4
86	ADD M	[A]←[A]+[[HL]]	***	7
C6	ADI DATA[8]	[A]←[A]+DATA[8]	***	7
8F	ADC A	[A]←[A]+[A]+C	***	4
88	ADC B	[A]←[A]+[B]+C	***	4
89	ADC C	[A]←[A]+[C]+C	***	4
8A	ADC D	[A]←[A]+[D]+C	***	4
8B	ADC E	[A]←[A]+[E]+C	***	4
8C	ADC H	[A]←[A]+[H]+C	***	4
8D	ADC L	[A]←[A]+[L]+C	***	4
8E	ADC M	[A]←[A]+[[HL]]+C	***	7
CE	ACI DATA[8]	[A]←[A]+DATA[8]+C	***	7
97	SUB A	[A]←[A]-[A]	0 1 0 1 0	4
90	SUB B	[A]←[A]-[B]	***	4
91	SUB C	[A]←[A]-[C]	***	4
92	SUB D	[A]←[A]-[D]	***	4
93	SUB E	[A]←[A]-[E]	***	4
94	SUB H	[A]←[A]-[H]	***	4
95	SUB L	[A]←[A]-[L]	***	4
96	SUB M	[A]←[A]-[[HL]]	***	7
D6	SUI DATA[8]	[A]←[A]-DATA[8]	***	7
9F	SBB A	[A]←[A]-[A]-C	***	4
98	SBB B	[A]←[A]-[B]-C	***	4
99	SBB C	[A]←[A]-[C]-C	***	4
9A	SBB D	[A]←[A]-[D]-C	***	4
9B	SBB E	[A]←[A]-[E]-C	***	4
9C	SBB H	[A]←[A]-[H]-C	***	4
9D	SBB L	[A]←[A]-[L]-C	***	4
9E	SBB M	[A]←[A]-[[HL]]-C	***	7
DE	SBI DATA[8]	[A]←[A]-DATA[8]-C	***	7




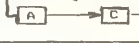
ША

- DD1 КР580ВМ80А
- DD2 КР580ГФ24
- DD3 К176 ИЕ5
- DD4 К155 ТМ2
- DD5, DD6 КР580 ИР82
- DD7 155 ИД3
- DD8 КР580 ИК38
- DD9, DD21 К155 ЛН1
- DD10, DD11 КР556 РТ5
- DD12, DD17 К155 ЛА8
- DD13, DD14 КР541Р42
- DD15, DD16 КР556 РТ4
- DD18, DD19 К155 РП1
- DD20 К155 ИД4
- DD22 К155 ЛЕ4
- DD23 К155 РЕ3
- DD24 К155 ИЕ5
- DA1 К140 УД6





Система команд микропроцессора

1	2	3	4	5
3C	INR A	[A] \leftarrow [A]+1	*** * -	5
04	INR B	[A] \leftarrow [B]+1	*** * -	5
0C	INR C	[A] \leftarrow [C]+1	*** * -	5
14	INR D	[A] \leftarrow [D]+1	*** * -	5
1C	INR E	[A] \leftarrow [E]+1	*** * -	5
24	INR H	[A] \leftarrow [H]+1	*** * -	5
2C	INR L	[A] \leftarrow [L]+1	*** * -	5
34	INR M	[[HL]] \leftarrow [[HL]]+1	*** * -	10
3D	DCR A	[A] \leftarrow [A]-1	*** * -	5
05	DCR B	[A] \leftarrow [B]-1	*** * -	5
0D	DCR C	[A] \leftarrow [C]-1	*** * -	5
15	DCR D	[A] \leftarrow [D]-1	*** * -	5
1D	DCR E	[A] \leftarrow [E]-1	*** * -	5
25	DCR H	[A] \leftarrow [H]-1	*** * -	5
2D	DCR L	[A] \leftarrow [L]-1	*** * -	5
35	DCR M	[[HL]] \leftarrow [[HL]]-1	*** * -	10
03	INX B	[DE] \leftarrow [DE]+1	--- --	5
13	INX D	[DE] \leftarrow [DE]+1	--- --	5
23	INX H	[HL] \leftarrow [HL]+1	--- --	5
33	INX SP	[SP] \leftarrow [SP]+1	--- --	5
0B	DCX B	[BC] \leftarrow [BC]-1	--- --	5
1B	DCX D	[DE] \leftarrow [DE]-1	--- --	5
2B	DCX H	[HL] \leftarrow [HL]-1	--- --	5
3B	DCX SP	[SP] \leftarrow [SP]-1	--- --	5
09	DAD B	[HL] \leftarrow [HL]+[BC]	--- -- * 10	10
19	DAD D	[HL] \leftarrow [HL]+[DE]	--- -- * 10	10
29	DAD H	[HL] \leftarrow [HL]+[HL]	--- -- * 10	10
39	DAD SP	[HL] \leftarrow [HL]+[SP]	--- -- * 10	10
*27	DAA		*** * *	4
A7	ANA A	[A] \leftarrow [A]&[A]	*** * 0	4
A0	ANA B	[A] \leftarrow [A]&[B]	*** * 0	4
A1	ANA C	[A] \leftarrow [A]&[C]	*** * 0	4
A2	ANA D	[A] \leftarrow [A]&[D]	*** * 0	4
A3	ANA E	[A] \leftarrow [A]&[E]	*** * 0	4
A4	ANA H	[A] \leftarrow [A]&[H]	*** * 0	4
A5	ANA L	[A] \leftarrow [A]&[L]	*** * 0	4
A6	ANA M	[A] \leftarrow [A]&[[HL]]	*** * 0	7
E6	ANI DATA[B]	[A] \leftarrow [A]&DATA[B]	*** * 0	7
AF	XRA A	[A] \leftarrow [A]X[A]	* 1 0 1 0	4
AB	XRA B	[A] \leftarrow [A]X[B]	*** 0 * 0	4
A9	XRA C	[A] \leftarrow [A]X[C]	*** 0 * 0	4
AA	XRA D	[A] \leftarrow [A]X[D]	*** 0 * 0	4
AB	XRA E	[A] \leftarrow [A]X[E]	*** 0 * 0	4
AC	XRA H	[A] \leftarrow [A]X[H]	*** 0 * 0	4
AD	XRA L	[A] \leftarrow [A]X[L]	*** 0 * 0	4
A6	ANA M	[A] \leftarrow [A]X[[HL]]	*** 0 * 0	7
EE	XRI DATA[B]	[A] \leftarrow [A]XDATA[B]	*** 0 * 0	7
B7	ORA A	[A] \leftarrow [A]V[A]	*** 0 * 0	4
B0	ORA B	[A] \leftarrow [A]V[B]	*** 0 * 0	4
B1	ORA C	[A] \leftarrow [A]V[C]	*** 0 * 0	4
B2	ORA D	[A] \leftarrow [A]V[D]	*** 0 * 0	4
B3	ORA E	[A] \leftarrow [A]V[E]	*** 0 * 0	4
B4	ORA H	[A] \leftarrow [A]V[H]	*** 0 * 0	4
B5	ORA L	[A] \leftarrow [A]V[L]	*** 0 * 0	4
B6	ORA M	[[HL]] \leftarrow [[A]V[[HL]]	*** 0 * 0	7
F6	ORI DATA[B]	[A] \leftarrow [A]VDATA[B]	*** 0 * 0	7
BF	CMP A	[A] - [A]	0 1 0 1 0	4
B8	CMP B	[A] - [B]	*** * * 0	4
B9	CMP C	[A] - [C]	*** * * 0	4
BA	CMP D	[A] - [D]	*** * * 0	4
BB	CMP E	[A] - [E]	*** * * 0	4
BC	CMP H	[A] - [H]	*** * * 0	4
BD	CMP L	[A] - [L]	*** * * 0	4
BE	CMP M	[A] - [[HL]]	*** * * 0	7
FE	CPI DATA[B]	[A] - DATA[B]	*** * * 0	7
2F	CMA	[A] \leftarrow [A]	--- --	4
07	RLC		--- -- * 4	4
0F	RRC		--- -- * 4	4
17	RAL		--- -- * 4	4
1F	RAR		--- -- * 4	4

		Команды ветвления		
C3	JMP ADDR[16]	[PC] \leftarrow ADDR[16]	--- --	10
C2	JNZ ADDR[16]	Z=0 JMP ADDR[16]	--- --	10
CA	JZ ADDR[16]	Z=0 JMP ADDR[16]	--- --	10
D2	JNC ADDR[16]	C=0 JMP ADDR[16]	--- --	10
DA	JC ADDR[16]	C=1 JMP ADDR[16]	--- --	10
E2	JPO ADDR[16]	P=0 JMP ADDR[16]	--- --	10
EA	JPE ADDR[16]	P=1 JMP ADDR[16]	--- --	10
F2	JM ADDR[16]	S=0 JMP ADDR[16]	--- --	10
FA	JM ADDR[16]	S=1 JMP ADDR[16]	--- --	10
CD	CALL ADDR[16]	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow ADDR[16]	--- --	17
CA	CNZ ADDR[16]	Z=0 CALL ADDR[16]	--- --	11/17
CC	CZ ADDR[16]	Z=1 CALL ADDR[16]	--- --	11/17
D4	CNC ADDR[16]	C=0 CALL ADDR[16]	--- --	11/17
DC	CC ADDR[16]	C=1 CALL ADDR[16]	--- --	11/17
E4	CP ADDR[16]	P=0 CALL ADDR[16]	--- --	11/17
EC	CPE ADDR[16]	P=1 CALL ADDR[16]	--- --	11/17
FA	CM ADDR[16]	S=0 CALL ADDR[16]	--- --	11/17
FC	CP ADDR[16]	S=1 CALL ADDR[16]	--- --	11/17
C9	RET	[[PC] \leftarrow [[SP]] [[PC] \leftarrow [[SP]+1 [[SP] \leftarrow [[SP]+2	--- --	10
C0	RNZ	Z=0 RET	--- --	5/11
C8	RZ	Z=1 RET	--- --	5/11
D0	RNC	C=0 RET	--- --	5/11
D8	RC	C=1 RET	--- --	5/11
E0	RPO	P=0 RET	--- --	5/11
E8	RPE	P=1 RET	--- --	5/11
F0	RP	S=0 RET	--- --	5/11
F8	RM	S=1 RET	--- --	5/11
E9	PCHL	[PC] \leftarrow [[HL]]	--- --	5
C7	RST 0	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0000H	--- --	11
CF	RST 1	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0008H	--- --	11
D7	RST 2	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0010H	--- --	11
DF	RST 3	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0018H	--- --	11
E7	RST 4	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0020H	--- --	11
EF	RST 5	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0028H	--- --	11
F7	RST 6	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0030H	--- --	11
FF	RST 7	[[SP]-1] \leftarrow [[PC]] [[SP]-2] \leftarrow [[PC]] [[SP] \leftarrow [[SP]-2 [PC] \leftarrow 0038H	--- --	11
Специальные команды				
FB	EI	разрешить прерывание	--- --	4
F3	DI	запретить прерывание	--- --	4
76	HLT	остановить работу	--- --	4
00	NOP	пустая команда	--- --	4
37	STC	C \leftarrow 1	--- --	1 4
3F	CMC	C \leftarrow \bar{C}	--- --	* 4

мо нажать на кнопку «8», а затем ввести два параметра — начальный и конечный адрес области памяти, в которой находится контрольная сумма. Так, можно проверить ПЗУ, если задать начальный и конечный адрес МОНИТОРА. Контрольная сумма выводится на индикаторы HL и для МОНИТОРА составляет 0181H(0000H—01FFH) и 18ССН(0200H—03FFH).

Теперь рассмотрим директиву ввода-вывода информации при работе с кассетным магнитофоном. Желательно, чтобы магнитофон имел счетчик ленты, по которому можно было бы найти начало записи. Идентифицировать же запись можно по выводимой при записи и чтении на индикатор HL контрольной суммы информации. Это как бы паспорт записи.

При записи данных на ленту нажимают кнопку «9», а затем вводят два параметра — начальный и конечный адреса данных. После набора директивы перед вводом младшего байта конечного адреса включают на запись магнитофон и лишь затем вводят последний байт адреса. После этого начнется запись информации на ленту. Скорость записи фиксированная — 1500 бит/сек. При выполнении этой директивы на ленту последовательно записываются 256 байт 00, ЕВ (байт синхронизации), младший байт начального адреса, старший байт начального адреса, младший байт конечного адреса и старший байт конечного адреса. Завершение выполнения директивы индицируется появлением на индикаторе HL контрольной суммы.

Чтение данных с магнитной ленты в ОЗУ производится по директиве «А». Директива имеет один параметр — смещение. Если смещение равно 0000, то программа считается в адреса, из которых она была записана. Если запись считана верно, то на индикаторах HL выводится контрольная сумма, которая индицировалась при записи. Итак, чтение записи с магнитофона начинают нажатием кнопки «А» и вводом смещения. Перед вводом последнего полубайта смещения магнитофон выключается на воспроизведение, и с началом записи (однотонное звучание) вводится последний полубайт смещения. Считывание записи завершается индикацией контрольной суммы данных, введенных в ОЗУ.

Директив запуска программ две. С помощью директивы «6» осуществляется запуск программы с адреса С000H, а с помощью директивы «7» программа может быть запущена с адреса, который вводится как параметр директивы. Нажав кнопку «7», набирают адрес пуска программы, контролируя ввод побайтно на индикаторе А. При правильном вводе адрес пуска выводится на индикатор HL. Затем можно вводить параметры запущенной программы. Пользуясь директивой «8» можно применять вспомогательные подпрограммы, входящие во вторую половину МОНИТОРА от 0200H до 03FFH.

С адреса 0200H запускается программа копирования данных. Ее параметрами являются начальный и конечный адрес исходной программы, а также начальный адрес копии. Копирование с помощью этой программы производится как вверх, так и вниз относительно исходной программы и даже на перекрывающиеся области программ с затиранием исходной программы в перекрывающихся областях. Предположим, вам необходимо скопировать МОНИТОР с адреса 0100H до 0120H в новые адреса ОЗУ с С100H до С120H. В этом случае должна соблюдаться следующая последовательность нажатия клавиш:

7 0200 0100 0120 С100

Копирование завершается с появлением двух единиц на индикаторе А. Это значит, что МОНИТОР ожидает ввода следующей директивы.

Проверить копирование данных можно с помощью другой вспомогательной программы сравнения данных двух областей. Ее пусковой адрес 03В2H. У этой программы три параметра: начальный и конечный адреса исходной программы и начальный адрес копии. В продолжение примера с копированием МОНИТОРА приведем последовательность нажатий на клавиши:

7 03B2 0100 0120 С100

При совпадении исходной программы и копии на всех индикаторах появляются единицы. При несовпадении эталона и копии высвечивается адрес с ошибкой копии. При этом возможно исправление ошибки и последующее сравнение данных.

Чтобы копированная программа могла быть запущена в новых адресах, ее нужно скорректировать с помощью вспомогательной программы-корректора. Ее пусковой адрес 025FH. У программы три параметра: начальный и конечный адреса исходной рабочей программы и начальный адрес корректируемой программы. Для нашего примера с МОНИТОРОМ это будет выглядеть так:

7 025F 0100 0120 С100

Завершается работа программы появлением единиц на индикаторе А. Если скорректированная программа будет использоваться на другом компьютере, например, «МИКРО-80» в адресном пространстве, которое отсутствует в микроЭВМ минимальной конфигурации, для подготовки такой программы используется суперкорректор, запускаемый с адреса 02E5H. Параметры суперкорректора: начальный, конечный адрес рабочей программы и начальный адрес подготавливаемой программы. Пример с МОНИТОРОМ — рабочая программа подготавливается для работы в F100—F120H:

7 02E5 С100 С120 F100

Завершается работа программы появлением единиц на индикаторе А. При отладке программ может возникнуть необходимость вставки или удаления отдельных байтов. Для этого можно использовать несколько программ.

Программа вставки байта — пусковой адрес 035EH. Параметры программы: адрес того места программы, где необходимо вставить байт, и адрес конца программы. Программа удаления байта — пусковой адрес 0388H. Параметры программы: адрес удаляемого байта и адрес конца программы. Программа завершается просмотром оставшейся части программы после адреса удаленного байта.

Программа замены адресов — пусковой адрес 0309H. После пуска программы вводятся начальный и конечный адреса рабочей программы, а затем старый и новый адреса. Завершается исполнение программы выходом в МОНИТОР и появлением единиц на индикаторе А.

Завершим описание вспомогательных программ МОНИТОРА программой индикации регистров микропроцессора. Эта программа может использоваться при отладке, для чего в отлаживаемую программу с помощью программы вставки байта производится вставка трех байт СD D0D3. После пуска отлаживаемой программы в требуемом месте программы происходит останов, а на индикаторах отображается содержимое регистров HL и ячейки памяти с адресом HL. При нажатии на любую клавишу происходит индикация остальных регистров, обозначение которых выводится на индикатор А в следующей последовательности: AF, BC, DE. Затем происходит исполнение отлаживаемой программы.

При нажатии кнопки «3» запускается директива тестирования дисплея микроЭВМ. После пуска директивы на всех индикаторах дисплея выводятся последовательно все шестнадцатеричные цифры от 0 до F.

Как уже отмечалось, в МОНИТОР встроена программа часов, работающая в режиме прерывания, что позволяет привязывать программы к реальному времени.

Сначала расскажем, что такое прерывание в микроЭВМ. Рассмотренные нами директивы МОНИТОРА исполняются в том порядке, в котором они расположены в памяти. Это значит, что никакое вмешательство пользователя не может воздействовать на ход выполнения программы, по которой уже начал работать процессор, кроме, конечно, сброса микроЭВМ в начальное положение. Система прерываний позволяет изменить ход выполнения программы на основании сигналов, поступающих в микропроцессор. После поступления прерывания выполнение основной программы прекращается и управление передается другой программе. Как только работа по другой программе завершится, будет осуществлен переход к прерванной программе и продолжится ее нормальное выполнение. Сигнал прерывания, поступающий в микропроцессор, асинхронный. В нашей микроЭВМ такими сигналами являются импульсы, поступающие каждую секунду от датчика времени, который выполнен на микросхеме К176ИЕ5, которая состоит из генератора импульсов частотой 32768 Гц и делителя с коэффициентом деления 2^{15} .

В процессе обработки прерывания, запрос на которое уже, предположим, поступил от датчика секунд, можно выделить следующие этапы обработки прерывания.

1. Вывод 14 микропроцессора КР580ВМ80А в переход в состояние логической 1, что означает наличие запроса на прерывание.

2. Микропроцессор принял запрос на прерывание, и на его выводе 16 установился уровень логического нуля.

Эти два события являются основными в рассматриваемой микроЭВМ.

Когда на выводе 16 устанавливается логический ноль, микропроцессор игнорирует все запросы на прерывание, поступающие на его вход. В системе команд микропроцессора есть две специальные команды: запрета прерываний DI и команда разрешения прерываний EI. С помощью этих команд можно менять состояние вывода 16 микропроцессора. После сброса микропроцессора в начальное состояние вывод 16 переводится в состояние логического 0. Это означает, что после начальной установки микропроцессор игнорирует все запросы на прерывание. Единственный способ разрешить прерывание — это заставить микропроцессор выполнить команду EI.

После выполнения этой команды на выводе 16 появится логическая 1. Именно поэтому после начальной установки одной из первых в МОНИТОРЕ выполняется команда EI, разрешающая прерывание. После этого первый же секундный импульс установит уровень логического 0 на выводе 16 и микропроцессор не будет воспринимать запросы на прерывание. В этот же момент формируется из слова состояния управляющий сигнал «Подтверждение прерывания» — ППР. Однако поскольку не одно устройство ввода-вывода и памяти не активизировано этим сигналом, то на шин данных будет действовать код команды FFH (команда RST7) и определит адрес перехода 0038H (подпрограмма прерывания) — программу часов. В нашей микроЭВМ организовано лишь одноуровневое прерывание по команде RST7.

Итак, по адресу 0038H осуществляется переход на подпрограмму часов, в которой программно организованы счетчики секунд, минут и часов. Ячейки памяти, в которых организованы эти счетчики, имеют адреса С3FDH, С3FEH и С3FFH. Завершается программа часов командой EI, и на выводе 16 микропроцессора снова устанавливается логическая 1. Прерванная программа продолжает работу.

В МОНИТОРЕ имеется директива начальной установки времени. После нажатия кнопки «С» прерывания запрещаются. Затем следует набрать адрес ячейки памяти С3FDH и последова-

тельно ввести значение секунд, минут и часов. После этого по сигналам точного времени нажать кнопку «УСТ», пустив тем самым часы, разрешив прерывания.

Имеется в МОНИТОРЕ и директива индикации времени, для чего достаточно нажать кнопку «В». При этом на индикаторе HL выводятся часы и минуты, а на индикаторе А — секунды. Точность хода часов определяется стабильностью кварцевого генератора. В состав МОНИТОРА входит ряд подпрограмм ввода-вывода, которые могут быть использованы при составлении программ. Перечислим эти программы и правила обращения к ним.

1. Подпрограмма записи байта на магнитофон, обращение с помощью команды RST1 (код CF). Код выводимого символа должен храниться в регистре «А».

2. Подпрограмма ввода байта с клавиатуры — обращение с помощью команды RST2 (код D7). После возврата из подпрограммы код клавиатуры находится в регистре «А» микропроцессора.

3. Подпрограмма задержки на секунду — обращение с помощью команд RST3 (код DF).

4. Подпрограмма чтения клавиатуры (проверка активности клавиатуры) — обращение через RST4 (код E7).

5. Подпрограмма индикации содержимого регистров HL и А на индикаторах дисплея микроЭВМ — обращение через команду RST5 (код EF).

6. Подпрограмма ввода двух байтов с клавиатуры в регистр DE — обращение через команду RST6 (код F7).

7. Подпрограммы сравнения адресов, хранимых в регистровых парах DE и BC поковой адрес 0194H и в регистровых парах HL и DE, пусковой адрес 0259H.

8. Подпрограмма чтения байта с магнитофона — пусковой адрес 0128H. После возврата из подпрограммы код считанного символа находится в регистре «А» микропроцессора.

Эти подпрограммы помогут вам упростить разработку новых программ. Заметим, что для унификации обмена программ с «МИКРО-80» подпрограммы обслуживания кассетного магнитофона (записи и чтения байта) идентичны тем, что использованы в МОНИТОРЕ «МИКРО-80».

Рассмотрим теперь пример, поясняющий применение приведенных подпрограмм. Предположим, что необходимо написать программу занесения констант в заданную область памяти. Входным параметром этой программы являются начальный и конечный адреса области памяти, в которую заносится константа. Алгоритм этой программы представлен в виде структурной схемы на рисунке 4.

При написании программы удобно составить бланк (листинг), позволяющий представить программу в трех ракурсах: колонка адресов, колонка объектных кодов, колонка меток и мнемонических (ассемблерных) кодов, однозначно определяющих работу программы на уровне команд, и колонка комментариев. Подобная форма записи программ позволяет не только понять работу программы (колонка комментариев), но и разобраться в структуре программы (колонка меток и операций). Колонка кодов непосредственно может быть введена в ОЗУ микроЭВМ. Такой бланк изображен ниже:

Адреса	Коды	Метки	Мнемоника	Комментарий
C000H	F7		RST6	ввод начального адреса в регистр DE
C001H	EB		XCHG	пересылка DE в HL
C002H	F7		RST6	ввод конечного адреса в регистр DE
C003H	D7		RST2	ввод константы в регистр А
C004H	47		MOV B,A	пересылка константы из А в В
C005H	70	M:	MOV M,B	пересылка константы из В в ОЗУ
C006H	CD		CALL 0259	обращение к п/программе сравнения адресов в HL и DE МОНИТОРА
C007H	59			
C008H	02			
C009H	23		INX H	увеличение адреса на единицу HL = HL+1
C00AH	C2		JNZM	проверка условия, если условие не выполняется, то переход на метку M
C00BH	05			
C00CH	CO			
C00DH	C7		RST0	выход в МОНИТОР по адресу 0000H

Для ввода кодов программы следует воспользоваться директивой «I». Затем проверить ввод директивой «2» и наконец осуществить пуск программы директивой «6». После нажатия кнопки «6» необходимо ввести начальный и конечный адрес ОЗУ, затем ввести константу. Окончание работы программы индицируется появлением единичек на правом индикаторе.

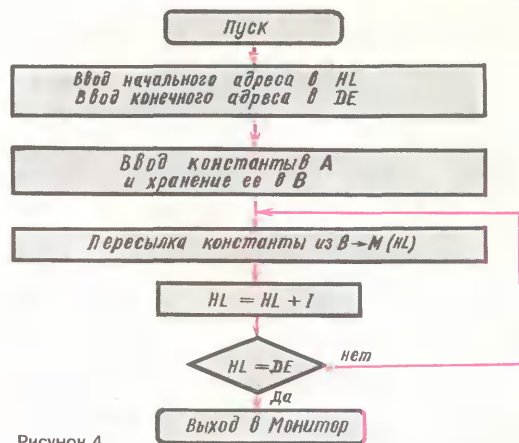


Рисунок 4.

Проверить результат работы программы можно с помощью директивы «5», посмотрев ту область памяти, куда была занесена константа. Затем эту программу можно для дальнейшего использования сохранить, записав на ленту.

Контрольная сумма этой программы 0800H.

В качестве еще одного примера приведем игровую программу «Лабиринт». Расположение этой программы приведена ниже. Дадим некоторые пояснения. «Лабиринт» — это поле 16x16 клеток, заполненное препятствиями. Координаты препятствий вводятся в виде отдельного набора данных с адреса C200H. Вариант расположения препятствий приведен в виде раскладки игрового поля.

Смысл игры заключается в следующем. Необходимо, мысленно представив поле размерами 16x16, пройти кратчайшим путем от нижнего правого угла (координаты FFH) к верхнему левому углу (00H). Управление движением осуществляется с помощью четырех кнопок «1» — движение на одну клетку влево; «2» — движение на одну клетку вправо; «3» — движение на одну клетку вверх; «4» — движение на одну клетку вниз.

Каждый ход сопровождается коротким звуковым сигналом. Если на пути встречается препятствие, то сигнала не будет, но шаг засчитывается. Число шагов выводится на индикаторы HL, а координаты движения выводятся на индикатор А. Сигнализация о правильности хода в лабиринте производится с помощью магнитофона, включенного на запись (с остановленной лентой). Пуск программы осуществляется директивой «6» с последующим нажатием на любую клавишу.

СБОРКА И НАЛАЖИВАНИЕ МИКРОЭВМ

МикроЭВМ собрана на плате из фольгированного стеклотекстолита. Расположение микросхем показано на рисунке 5. Общий провод и цепь питания +5В выполнены печатным способом проводниками максимальной ширины, а остальные монтажные соединения — тонким изолированным проводом. В местах подключения печатных проводников к выводам питания микросхем подключаются блокировочные конденсаторы емкостью 0,22—0,68 МкФ. Для уменьшения паразитных связей и наводок монтажные провода надо прокладывать по кратчайшим путям между соседними соединяемыми выводами микросхем и не связывать в жгут. Удобно использовать обмоточный провод марки ПЭПЛОТ или ПЭВТЛК. Для соединения микроЭВМ с периферийными устройствами установите на плате разъем типа СНП34 или контактные штыри. Контактные штыри, подключенные к печатным проводникам платы общего провода, и +5В соединяются с клеммами источника питания проводами большого сечения.

В микроЭВМ можно применить любую клавиатуру, в которой каждая клавиша связана с контактами, работающими на замыкание. Можно, например, воспользоваться клавиатурой от микрокалькулятора. Возможно размещение клавиатуры на плате микроЭВМ. Для уменьшения количества соединительных проводов индикаторы дисплея постарайтесь разместить на монтажной плате. Возможны и другие варианты конструкции. Указанные на принципиальной схеме интегральные микросхемы серии K155 можно заменить их функциональными аналогами се-

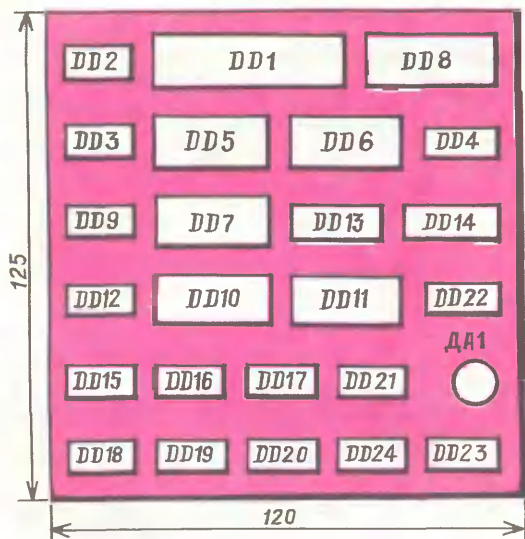


Рисунок 5.

рий K133, K555, K531. Микросхемы буфера шины адреса КР580ИР82 можно заменить микросхемами К589ИР12 или двумя микросхемами К589АП16. ОЗУ можно собрать практически на любых статических микросхемах памяти. Если применяемые микросхемы ОЗУ имеют раздельные вход и выход, то эти выводы соединяют вместе.

Хотя наш первый модуль и отличается предельной простотой, однако наличие в нем тесной взаимосвязи между аппаратными средствами и программным обеспечением приводит к тому, что даже незначительная неисправность в аппаратуре или в программном обеспечении приводит к неработоспособности микроЭВМ.

Начать отладку микроЭВМ целесообразно с проверки омметром всех связей на монтажной плате и устранения выявленных дефектов монтажа. С особой тщательностью следует проверить наличие электрических связей между выводами питания микросхем и контактами источника питания. Щупами омметра касайтесь непосредственно выводов микросхем — это поможет вам обнаружить дефекты пайки. Затем проверяют наличие питающих напряжений на выводах микропроцессора и на других микросхемах модуля.

Следующий этап — проверка схемы формирования синхроимпульсов. Для этого вам потребуется осциллограф. Убедившись в наличии синхроимпульсов Ф1 и Ф2 на выводах микропроцессора, переходят к проверке функционирования блока центрального процессора при постоянно действующей на шине данных команде NOP (00H). Код команды 00 принудительно поддают на шину данных с помощью переключек.

При этом на адресной шине должен происходить последовательный перебор всех адресов, который можно проконтролировать с помощью осциллографа. Сигналы на адресной шине в этом случае имеют форму симметричных прямоугольных импульсов, причем частота импульсов уменьшается вдвое при увеличении номера разряда на единицу. При правильной работе команды NOP переходят к проверке остальных блоков микроЭВМ. Для этого вывод 3 микросхемы DD2 соединяют с общим проводом и убеждаются в наличии высокого уровня на выводе 24 микропроцессора, что свидетельствует о нахождении его в состоянии ожидания. Нажав на кнопку «УСТ» проверяют на всех линиях шины адреса нулевого уровня, а на линиях шины данных двойного кода 0011 0001, записанного в нулевой ячейке МОНИТОРА (микросхема ПЗУ DD10).

Имитировать сигнал выборки индикатора можно, отключив провод от вывода 10 микросхемы DD12.3 и соединив его с общим проводом. На правом крайнем индикаторе появится код 31. Затем отключите провод от вывода 13 микросхем DD18 и DD19 — код 31 появится уже на среднем индикаторе. Припаяйте на место провод к выводу 13 и, отключив провод от вывода 14 микросхем DD18 и DD19, проследите за индикацией кода 31 на левом индикаторе. Затем припаяйте на место все отключенные от микросхем DD18 и DD19 провода, а микросхему DD10 выньте из панели или отключите провод от ее 20-го вывода.

Отключив провод от вывода 8 микросхемы DD22, имитируют

подачу сигнала выборки клавиатуры. Нажимая на различные клавиши, убеждаются в индикации на правом индикаторе кодов нажатых клавиш. Восстанавливают провод, подключенный к выводу 8 микросхемы DD22. Снимают переключку, подключенную к выводу 3 микросхемы DD1. Вставляют в панельку микросхему ПЗУ DD10. Собирают на отдельной плате узел (рис. 6), с помощью которого проверяют поцикловое исполнение программы МОНИТОР микропроцессором. При однократном нажатии на кнопку «ШАГ» микропроцессор переходит в состояние ожидания, что позволяет на индикаторах дисплея контролировать считываемые из ПЗУ коды и их исполнение.

Для исключения влияния сигнала прерывания провод от вывода 14 микропроцессора (DD1) отключают.

Нажав кнопку «УСТ» и последовательно нажимая кнопку «ШАГ», контролируют выполнение первых команд МОНИТОРА. В частности, после девяти нажатий на правом индикаторе должны появиться две единицы, при этом на адресной шине должен быть код адреса 9000H, а на шине управления активен сигнал ПЗУ. При обнаружении несоответствий следует проверить, нет ли замыканий между линиями шины адреса, или шины данных, или другими сигнальными линиями.

После устранения обнаруженных неисправностей, добившись исполнения первых команд МОНИТОРА в шаговом режиме, отключают дополнительный узел. Подключают вывод 10 микросхемы DD12.3 к соответствующему проводу, отключив его от общего.

Переходят к проверке микроЭВМ в динамике. Нажав на кнопку «УСТ», убеждаются в появлении двух единиц на правом индикаторе. Затем нажимают клавишу «>» и проверяют прохожденье теста индикации.

Нажав на клавишу «4», проверяют ОЗУ, затем проверяют работу директив с памятью, последовательно проверяя запись данных в ОЗУ с адреса С000H (директива «1»), чтение данных из ОЗУ (директива «2»), запись данных в ОЗУ с любого адреса (директива «В») и чтение данных из ОЗУ с любого адреса (директива «5»). Если все эти директивы не выполняются, то набирают программу заполнения константой (она вводится с клавиатуры) адресного пространства, задаваемого также с клавиатуры.

Проверяют с помощью этой программы исполнение директивы пуска программы с любого адреса.

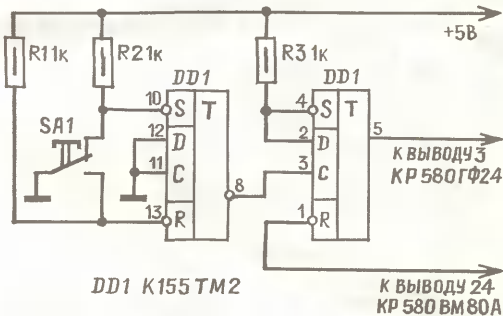


Рисунок 6.

Восстановив провод, идущий к выводу 14 микропроцессора и нажав клавишу «В», проверяют индикацию времени, смену показаний на правом индикаторе каждую секунду. Проверяют директиву установки времени «С». Пуск часов производится по сигналам точного времени нажатием кнопки «УСТ». Затем переходят к проверке директив работы с кассетным магнитофоном.

Нажав на клавишу «9», а затем набрав адреса D000H-FFFFH, наблюдают с помощью осциллографа непрерывную последовательность импульсов на выводе 9 микросхемы DD4.2. При этом тон фонограммы, которая записана на магнитофон, должен быть все время одним и тем же. Записанную на магнитофон информацию используют для проверки работы формирователя импульсов на микросхеме DA1.

Включив магнитофон на воспроизведение, проверяют с помощью осциллографа на выводе 6 микросхемы DA1 прямоугольную последовательность импульсов.

Затем проверяют запись и чтение программы занесения константы в задаваемое адресное пространство. Конечно, не всегда проверка микроЭВМ требует таких больших усилий. Как показывает опыт, если все детали исправны и при монтаже не было допущено ошибок, то микроЭВМ начинает работать сразу после сборки.

Для того чтобы вы могли убедиться в том, что собранный вами компьютер работает, мы предлагаем вам на выбор несколько простых, но увлекательных игровых программ:

1. Программа «Реакция» загружается директивой «1» с C000H по C05DH. После пуска программы директивой «б» нажимают клавишу «0», затем испытуемый нажимает любую клавишу, при этом на всех индикаторах дисплея появляются «FF». Сразу после того, как цифры на индикаторах начинают бежать, испытуемый должен нажать на кнопку «УСТ». Индивидуальный код на индикаторах и характеризует вашу реакцию. Чем он меньше, тем реакция быстрее.

```
C000  07 .F1 .00 CA 27 .C5 .6 .7 .8 .9 .A .B .C .D .E .F
C010  FE 03 CA 31 C0 FE 04 CA 36 C0 FE 05 CA 3B C0 C3
C020  40 C0 06 0B C3 41 C0 06 0E C3 41 C0 06 07 C3 41
C030  C0 06 0F C3 41 C0 06 10 C3 41 C0 06 0D C3 41 C0
C040  C0 06 0F 3E FF 21 FF FF FF DF 05 C2 47 C0 3E C0 21
C050  00 00 EF 23 00 00 C3 51 C0 00 C0 C3 10 C0 FF FF
```

2. Программа «Гамма» загружается директивой «1» с C000H по C082H. Пуск программы осуществляют директивой «У» с адреса C03EH. После ввода пускового адреса необходимо ввести длительность звучания нот и интервал пауз между нотами в виде двухразрядных шестнадцатеричных цифр в диапазоне от 01 — FFH. Прослушивают гамму, подключив вход магнитофона, включенного на запись к разъему интерфейса.

```
  .0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .A .B .C .D .E .F
C000  00 00 00 D5 1E 01 21 C9 0C 28 7B D3 A1 C0 1D C0 2F
C010  5F 00 7C BA C2 09 C0 15 C2 06 C0 19 C9 41 2B 05
C020  C2 1E C0 09 21 CA 0C 2B 7D BA C2 27 C0 C9 05 21
C030  CA 0C 2B 7D BA C2 32 C0 1D C2 2F C0 D1 C9 F7 0E
C040  8B CD 03 C0 CD 2E C0 0E 7A CD 03 C0 CD 2E C0 0E
C050  6C CD 03 C0 CD 2E C0 0E 65 CD 03 C0 CD 2E C0 0E
C060  5A CD 03 C0 CD 2E C0 0E 4F 03 CD 03 C0 CD 2E C0
C070  0E 47 CD 03 C0 CD 2E C0 0E 44 CD 03 C0 CD 2E C0
C080  C3 3F C0 00 00 00 FF FF FF FF FF FF FF FF FF
```

3. Программа «Крестик-нолики» загружается директивой «1» с C000H по C065H. Поле игры в виде 9 клеток пронумеровано так, как показано на схеме. После пуска программы директивой «0» на крайнем справа индикаторе появляется цифра «09» — это первый ход микроЭВМ, которая всегда ходит первая с центра. Место вашего нолика вы вводите с клавиатуры, указав номер клетки поля. Вы можете проиграть микроЭВМ, в этом случае на дисплее появляется код 73. В случае ничьей индицируется код «11».

```
  .0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .A .B .C .D .E .F
C000  3E 09 32 00 9A D7 CD 1A C0 0C 04 27 C0 00 00 00
C010  7A CD 1A C0 7A CD 1A C0 C7 04 C0 01 90 C2 22 C0
C020  3E 08 4F 32 00 90 D7 57 1E 0A 79 93 CA 32 C0 F2
C030  35 C0 2E 08 85 6F 7A 95 CA 45 C0 00 26 73 7C
C040  32 00 90 EF 76 79 C9 FE 07 CA 5E C0 FE 03 CA 5E
C050  C0 FE 05 CA 5E C0 FE 07 CA 5E C0 C3 10 C0 79 CD
C060  1A C0 90 C3 3D C0 FF FF FF FF FF FF FF FF FF
```

Нумерация клеток поля игры «Крестик-нолики».

01	02	03
08	09	04
07	06	05

Игровая программа «Лабиринт»

```
  .0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .A .B .C .D .E .F
C000  00 00 00 E7 C4 00 C2 0E C0 26 C2 C3 10 C0 26 C1
C010  2E FF 7D 32 F5 C0 32 F6 C0 42 C0 90 11 00 00 EF
C020  0E 7D CA 33 C0 FE 01 CA 32 C0 FE 02 CA 64 C0 C3
C030  52 C0 00 3A F6 C0 D6 10 32 F6 C0 DA 8E C0 C3 76
C040  C0 00 3A F6 C0 C6 10 32 F6 C0 DA 8E C0 C3 76
C050  C0 00 3A F6 C0 C6 01 32 F6 C0 CD C6 C0 DA 8E C0
C060  C3 76 C0 00 3A F6 C0 D6 01 32 F6 C0 CD C6 C0 DA
C070  8E C0 C3 76 C0 00 4F 7E C4 00 0F C4 00 00 00 00
C080  C0 32 F6 C0 32 00 90 CD D5 C0 C3 9C C0 00 3A F5
C090  C0 32 F6 C0 32 00 90 F5 C3 A5 C0 00 3A F5 C0 C6
C0A0  00 CA B7 C0 F5 7B 3C 27 5F D2 AF C0 7A 3C 57 F1
C0B0  E3 EF EB C3 20 C0 00 00 C0 DF 00 DB A0 C6 00
C0C0  CA B7 C0 C3 03 C0 C9 C1 79 E6 16 CA D1 C0 C7 78
C0D0  C9 37 3F 75 D5 E3 1E 01 16 50 21 FF 00 7B D3
C0E0  A1 C8 EF C0 2F 5F 2B 7C B5 C2 DE C0 E1 D1 C9 42
C0F0  05 C2 F0 C0 C9 D2 D2 21 C0 C2 3E FF EF E7 EF DF
```

Пример поля игры «Лабиринт»

```
  .0 .1 .2 .3 .4 .5 .6 .7 .8 .9 .A .B .C .D .E .F
C200  00 00 01 01 01 01 01 01 01 01 01 01 01 01 01
C210  00 00 00 00 00 00 01 00 01 00 00 00 00 00 01
C220  01 00 00 00 01 00 00 01 00 00 01 00 00 00 01
C230  01 00 01 00 00 00 00 00 00 01 00 00 01 00 01
C240  01 00 00 00 01 00 00 00 01 00 00 01 00 00 01
C250  01 00 01 00 00 00 00 00 01 00 00 01 00 00 01
C260  01 00 00 00 00 01 00 00 00 00 00 00 00 00 01
C270  01 00 00 00 00 00 00 00 00 01 00 00 01 00 01
C280  01 00 01 00 00 01 00 01 00 01 00 01 00 00 01
C290  01 00 00 00 00 00 00 01 00 00 00 00 00 00 01
C2A0  01 00 00 00 01 00 00 00 01 00 00 01 00 00 01
C2B0  01 01 00 00 00 00 01 00 00 00 00 00 00 00 01
C2C0  01 00 00 00 00 00 00 00 01 00 00 00 01 00 01
C2D0  01 00 01 00 00 01 00 00 00 00 01 00 01 00 01
C2E0  01 00 00 01 00 00 01 00 00 01 00 00 01 00 00
C2F0  01 01 01 01 01 01 01 01 01 01 01 01 01 01 00 00
```

СПРАВОЧНЫЙ ЛИСТОК «ЮТ-88»

Микросхема КР5808ВМ80А — функционально законченный однокристалльный микропроцессор с фиксированной системой команд. Применяется в устройствах обработки данных и управления.

Микропроцессор имеет раздельные 16-разрядный канал адреса и 8-разрядный канал данных. Канал адреса обеспечивает прямую адресацию к внешней памяти объемом до 65 536 байт и 256 устройствам ввода и вывода.

Условное графическое обозначение микросхемы познано на рисунке 1, название выводов указано в таблице 1. Структурная схема микропроцессора изображена на рисунке 2.

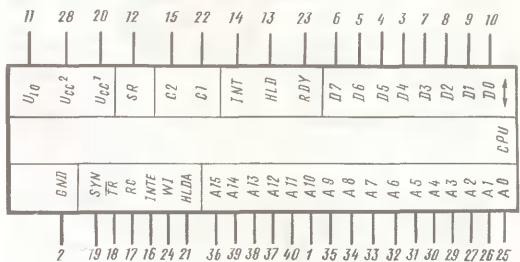


Рисунок 1.

Восьмиразрядное арифметико-логическое устройство микропроцессора выполняет арифметические и логические операции над двоичными числами, представленными в дополнительном коде, а также обработку двоично-десятичных упакованных чисел.

В состав блока регистров входят: 16-разрядный регистр адреса команды (IP), 16-разрядный регистр указателя стека (SP), 16-разрядный регистр временного хранения (WZ), 16-разрядная схема инкремента-декремента и шесть 8-разрядных регистров общего назначения (B, C, D, E, H, L), которые могут использоваться и как три 16-разрядных регистра (BC, DE, HL).

Микропроцессор выполняет команды по машинным циклам. Число циклов, необходимое для выполнения команды, зависит от ее типа и может быть от одного до пяти. Машинные циклы выполняются по машинным тактам. Число тактов в цикле определяется кодом выполняемой команды и может быть от трех до пяти. Длительность такта равна периоду тактовой частоты и при частоте 2,0 МГц составляет 500 наносекунд.

В начале каждого машинного цикла микропроцессор вырабатывает сигнал синхронизации SYN, который в сочетании с другими сигналами может быть использован для организации различных режимов работы процессора.

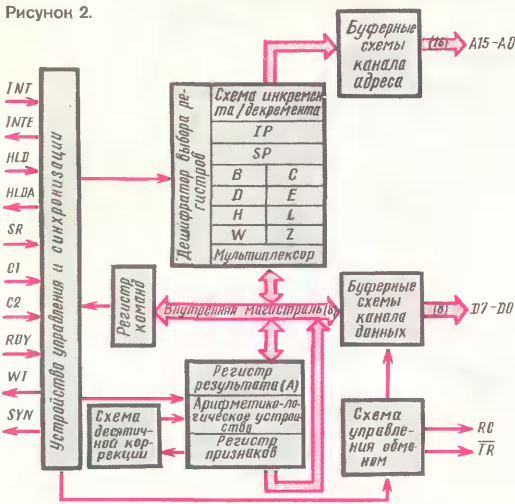
На рисунке 3 познана диаграмма состояний типичного машинного цикла, показывающая последовательность перехода от такта и такту в машинном цикле и влияние внешних сигналов RDY, HLD и INT на его выполнение. После подачи на вывод SR сигнала высокого уровня микропроцессор устанавливается в исходное состояние.

В такте T1 микропроцессор выдает на адресный канал адрес ячейки, в которой хранится команда программы, а через канал данных — информацию состояния. В такте T2 анализируются состояния сигналов на входе RDY, «Подтверждение останова», и в зависи-

Таблица 1

Вывод	Обозначение	Тип вывода	Функциональное назначение выводов
1, 25—27 29—40	A10, A0—A2 A3—A9, A15, A12—A14, A11	Выходы	Канал адреса
2	GND	—	Общий
3—10	D4—D7, D3—D0	Входы/выходы	Канал данных
11	U ₀	Вход	Напряжение источника смещения —5В
12	SR	Вход	Установка в исходное состояние
13	HLD	Вход	Захват
14	INT	Вход	Запрос прерывания
15, 22 16	CI, C2 INTE	Входы Выход	Тантовые сигналы Разрешение прерывания
17	RC	Выход	Прием информации
18	TR	Выход	Выдача информации
19	SYN	Выход	Сигнал синхронизации
20	Ucc1	—	Напряжение питания +5В
21	HLDA	Выход	Подтверждение захвата
23	RDY	Вход	Сигнал «Готовность»
24	W1	—	Сигнал «Ожидание»
28	Ucc2	—	Напряжение питания +12В

Рисунок 2.



мости от состояния этих сигналов микропроцессор переходит в состояние ожидания, останова или приступает к выполнению такта T3. В такте T3 при наличии сигнала высокого уровня на входе RDY микропроцессор принимает информацию по каналу данных, анализирует состояние сигнала на входе HLD, и если этот сигнал высокого уровня, то после окончания такта T3 переходит в состояние захвата. В зависимости от кода выполняемой команды машинный цикл завершается после выполнения тактов T3, T4 или T5.

В конце машинного цикла снова анализируется состояние сигнала на входе HLD. При низком уровне сигнала проверяется, окончено ли выполнение команды. Если команда не закончена, то микропроцессор выполняет следующий машинный цикл команды, начиная с такта T1. В конце каждой команды микропроцессор анализирует состояние сигнала на входе INT. Если сигнал высокого уровня и прерывание ранее было разрешено командой EI, то микропроцессор переходит к выполнению машинного цикла «Прерывание», начиная с такта T1. В противном случае первый машинный цикл новой команды начинается с такта T1.

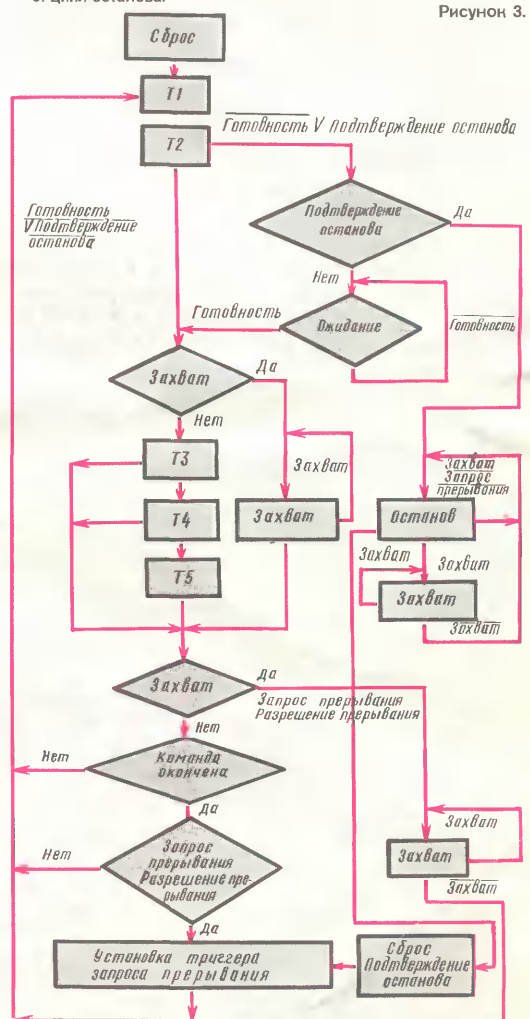
Действия, выполняемые микропроцессором в конкретном машинном цикле, определяются 8-разрядной информацией состояния, которая выдается через канал данных в такте T1 каждого машинного цикла. Эта информация может использоваться для выработки сигналов обращения к микросхемам памяти, устройствам ввода/вывода и для организации различных режимов работы микропроцессора.

Таблица 2

Разряд канала данных	Сигнал состояния	Цикл M1	Цикл чтения ЗУ	Цикл записи в ЗУ	Цикл чтения стека	Цикл записи в стек	Цикл ввода	Цикл вывода	Цикл прерывания	Цикл останова	Цикл прерывания при останове
D0	Подтверждение прерывания	0	0	0	0	0	0	0	1	0	1
D1	Запись/вывод	1	1	0	1	0	1	1	1	1	1
D2	Стек	0	0	0	1	1	0	0	0	0	0
D3	Подтверждение останова	0	0	0	0	0	0	0	0	1	1
D4	Вывод	0	0	0	0	0	0	1	0	0	0
D5	M1	1	0	0	0	0	0	0	0	0	1
D6	Ввод	0	0	0	0	0	1	0	0	0	0
D7	Чтение	1	1	0	1	0	0	0	0	1	0

- В зависимости от сочетания сигналов состояния, выдаваемых в конкретном цикле, машинные циклы можно разделить на 10 типов:
1. Цикл M1 — прием первого байта команды в регистр команд.
 2. Цикл чтения ЗУ (запоминающего устройства) по содержимому программного счетчика или содержимому одного из регистров BC, DE, HL.
 3. Цикл записи в ЗУ — запись в ЗУ по содержимому одного из регистров BC, DE, HL.
 4. Цикл чтения стека — чтение ЗУ по содержимому указателя стека.
 5. Цикл записи в стек — запись в ЗУ по содержимому указателя стека.
 6. Цикл ввода — ввод информации в регистр результата (анкумулятор) из внешнего устройства.
 7. Цикл вывода — вывод информации из регистра результата во внешнее устройство.
 8. Цикл прерывания — прием кода команды RST или CALL из контроллера прерываний.
 9. Цикл останова.

Рисунок 3.



10. Цикл прерывания при останове — прием кода команды RST или CALL при выводе микропроцессора из режима «Остановка» по прерыванию.

Наименования сигналов состояния, соответствие их разрядам канала данных и тактам машинных циклов приведены в таблице 2. При выполнении команд микропроцессор может переходить в одно из трех состояний: «ожидание», «захват» или «останов», длительность которых определяется внешними управляющими сигналами.

Сигнал высокого уровня на входе RDY обеспечивает автоматическое выполнение команд программы микропроцессором с частотой тактовых сигналов. Если на выводе RDY установлен сигнал низкого уровня, то микропроцессор переходит в режим «Ожидание» и формирует выходной сигнал W1 высокого уровня.

Сигнал RDY может быть использован для согласования работы микропроцессора с работой медленно действующих устройств, если длительность их цикла обращения составляет более одного периода тактовой частоты, а также для организации пошагового (по циклам) выполнения команд или покомандного выполнения программы.

При подаче на вход HLD сигнала высокого уровня микропроцессор переходит в состояние «захват» и подтверждает переход в это состояние формированием сигнала высокого уровня на выводе HLD.

Буферные схемы канала адреса и данных микропроцессора переключаются в высокоомное (высокоимпедансное) состояние, а выходные управляющие сигналы — в состояние низкого уровня (за исключением сигналов TR и HLD). Микропроцессор переходит в состояние «захват» в такте T3, если выполняется цикл чтения и на входе RDY сигнал высокого уровня, и в такте, следующем за T3, если выполняется цикл записи. Сигналы HLD и HDA позволяют организовать режим прямого доступа к памяти для любого внешнего устройства, формирующего сигнал HLD.

При выполнении команды HLT микропроцессор переходит в состояние «останов» и переводит буферные схемы канала адреса и данных в высокоомное состояние. Из состояния «останов» микропроцессор выходит при наличии сигнала высокого уровня на одном из его входов:

на входе SR — микропроцессор начинает работать с такта T1 цикла M1;

на входе HLD — микропроцессор переходит в состояние «захват», а после перехода сигнала HLD на низкий уровень возвращается в состояние «останов»;

на входе INT — микропроцессор переходит к выполнению цикла прерывания при остове с такта T1, если команде HLT предшествовала команда EI «разрешение прерывания», иначе он остается в состоянии «останов».

Сигнал высокого уровня на выводе INT позволяет прерывать выполнение текущей программы и переводит микропроцессор на выполнение подпрограммы обслуживания устройства, выдавшего запрос прерывания (например, часе). При поступлении сигнала INT микропроцессор (после окончания текущей команды) переходит с такта T1 к выполнению машинного цикла «Прерывание» в том случае, если прерывание было разрешено ранее командой EI. При выполнении цикла «Прерывание» в такте T1 микропроцессор выдает по шине данных сигнал состояния «Подтверждение прерывания». По окончании подпрограммы прерывания происходит возврат к прерванной программе.

Сигнал высокого уровня на входе SR (длительность которого должна быть не менее трех периодов тактовой частоты) устанавливает микропроцессор в исходное состояние: триггеры разрешения прерывания и захвата, регистры команд, признанов и адреса команды устанавливаются в нулевое состояние. После окончания действия сигнала SR микропроцессор производит первое обращение за чтением команды и ячейки памяти по адресу 0000H.

Система команд микропроцессора состоит из 78 базовых команд, которые можно разделить на пять групп:

- команды передачи данных — используются для передачи данных из регистра в регистр, из регистра в память;
- арифметические команды — используются для сложения, вычитания, инкремента или декремента содержимого регистров или ячейки памяти;
- логические команды — И, ИЛИ, сравнение, сдвиги;
- команды переходов — используются для условных или безусловных переходов, вызова подпрограммы и возврата из них;
- команды управления, ввода/вывода и работы со стекном — используются для управления регистром признанов, ввода и вывода информации.

В микропроцессоре KP580BM80A принят формат информационного слова, представляющего собой 8-разрядное двоичное слово (байт). Формат информационного слова (данных):

D7 D6 D5 D4 D3 D2 D1 D0

где D7 — старший разряд слова, а D0 — младший. Отрицательные числа хранятся в памяти в дополнительном коде.

Формат команды зависит от типа операции — может быть одно-,

двух- и трехбайтовым. Байты двух- и трехбайтовых команд должны храниться в ячейках памяти, следующих одна за другой. Адрес первого байта всегда является адресом мода операции. Формат команд микропроцессора показан ниже.

Однобайтовая команда

Адрес ячейки памяти	D7	D6	D5	D4	D3	D2	D1	D0	Код операции
---------------------	----	----	----	----	----	----	----	----	--------------

Двухбайтовая команда

Байт В1	D7	D6	D5	D4	D3	D2	D1	D0	Код операции
Байт В2	D7	D6	D5	D4	D3	D2	D1	D0	Данные или адрес порта

Трехбайтовая команда

Байт В1	D7	D6	D5	D4	D3	D2	D1	D0	Код операции
Байт В2	D7	D6	D5	D4	D3	D2	D1	D0	Данные
Байт В3	D7	D6	D5	D4	D3	D2	D1	D0	или адрес

Все команды микроЭВМ сведены в таблице (см. влчадку). Выделим наиболее общие закономерности их использования.

1. Арифметические и логические операции разрешены только между аккумулятором и байтом данных или между аккумулятором и любым регистром.

2. Аккумулятор и регистровая пара HL являются единственными регистрами, которые могут быть непосредственно загружены в память.

3. Аккумулятор является единственным регистром, который может быть инвертирован, сдвинут, всосанно загружен в память с использованием регистровых пар BC и DE или использован в командах ввода/вывода (IN, OUT).

4. Регистровая пара HL является единственной, содержимое которой может быть передано в счетчики команд (команда PCHL) или указатель стека (команда SPHL).

5. Регистровая пара H не может использоваться как аккумулятор двойной длины при сложении 16-разрядных чисел (команда DAD).

6. Содержимое регистровых пар HL и DE можно менять местами (команда HCHG).

7. Отдельные команды могут применяться для специальных функций. Команды XRAA или SUBA обнуляют аккумулятор. Команды ANA или ORA А очищают флаг переноса, а команда ADDA выполняет логический сдвиг аккумулятора влево.

8. Команды увеличения (уменьшения) на 1 регистров (INR, DCR) действуют на все флаги, за исключением флага переноса. Команды увеличения (уменьшения) на 1 регистровых пар (INX, DCX) не оказывают влияния на флаги.

9. В стек или из стека могут быть переданы только регистровые пары. Одной из таких регистровых пар является слово состояния микропроцессора, которое содержит аккумулятор (старший байт) и флаги (младший байт). Команды CALL и RET передают адрес в стек и обратно.

10. При записи 16-разрядных адресов младший байт записывается первым.

Операнды команд могут храниться в программно доступных регистрах микропроцессора или памяти. Для указания операнда в регистре используется регистровая и регистровая неявная адресации, для указания операнда в памяти — непосредственная, прямая, косвенная регистровая и стековая адресации.

Регистр признанов микропроцессора используется для хранения пяти битов признанов, которые вырабатываются в результате выполнения некоторых операций:

S — бит знака; равен 1, если старший значащий разряд результата операции равен 1 (то есть результат операции — отрицательное число);

Z — бит нуля; равен 1, если результат операции равен нулю;

AC — бит вспомогательного переноса; равен 1, если при выполнении операции был перенос из третьего разряда сумматора в четвертый;

C — бит переноса; равен 1, если при выполнении операции был перенос из седьмого разряда сумматора или заем в седьмой разряд сумматора;

P — бит четности; равен 1, если число единиц результата четное.

Распределение разрядов в регистре признанов следующее:

D7 D6 D5 D4 D3 D2 D1 D0

S Z AC 0 P 1 C



Главный редактор В. В. СУХОМЛИНОВ
 Редактор предложения В. А. ЗАБОРОТОВ
 Художественный редактор А. М. НАЗАРЕНКО
 Технический редактор Е. А. МАКСИМОВА

Сдано в набор 26.12.88. Подп. в печ. 17.01.89. А04627. Формат 60x90¹/₃. Печать офсетная. Бумага офсетная № 1. Условн. печ. л. 2. Условн. кр.-отг. л. 4. Учетно-изд. л. 2,5. Тираж 1 200 000 экз. Заказ 317. Цена 20 коп.

Типография ордена Трудового Красного Знамени ИПО ЦК ВЛКСМ «Молодая гвардия».

Адрес ИПО: 103030, Москва, К-30, Суцевская, 21.
 Адрес редакции: 125015, Москва, Новодмитровская, 5а. Тел.: 285-80-94
 Издательско-полиграфическое объединение ЦК ВЛКСМ «Молодая гвардия».