

Схемотехника больших систем

Микропроцессорные системы.

Развитие микроэлектроники, обеспечившее быстрый рост степени интеграции, микросхем, появление больших интегральных схем (БИС – от 1000 элементов на кристалле) и сверх - БИС (СБИС – свыше 10000 элементов на кристалле) поставило в начале 70-х годов новые проблемы перед разработчиками микросхем. С одной стороны, экономика производства требовала большого объема выпуска БИС при ограниченной их номенклатуре (числе типов). С другой стороны *каждый тип БИС, представляющих собой «жесткую», неизменяемую структуру из сотен и тысяч логических элементов, мог выполнять только одну определенную функцию или весьма ограниченный их набор*, т. е. БИС были узкоспециализированными. Поэтому число типов БИС, требуемых для реализации разнообразных цифровых систем, возрастало, а требуемый объем выпуска каждого из типов БИС оказывался небольшим.

Решение проблемы было достигнуто разработкой БИС, функции которых задаются путем подачи внешних электрических сигналов, изменяющихся по определенной программе. Такие программируемые БИС получили название *микропроцессоров*. Качественным отличием микропроцессоров от других типов микросхем является *возможность их функциональной перестройки с помощью изменения внешней программы*.

Структура и принцип работы цифровых систем.

Цифровые системы производят обработку информации путём выполнения над ней определенной последовательности операций, задаваемой соответствующим *алгоритмом*. Большие цифровые системы строятся исключительно по синхронному принципу работы.

Алгоритм – набор формальных правил, четко и однозначно определяющих процесс выполнения заданной работы.

Работа цифровой микропроцессорной системы основана на *принципе программного управления* обработкой информации.

Программа – формализованное описание реализуемого в системе **алгоритма** обработки информации, данное в виде последовательности управляющих команд.

Команда – двоичный код, задающий действия системы по выполнению конкретной операции.

Микропроцессорные системы являются типичным представителем больших цифровых систем. Основой такой системы является микропроцессор.

Процессор – устройство обработки информации, осуществляемой по программе.

Микропроцессор (МП)– процессор выполненный по интегральной технологии в одной или нескольких БИС. Набор команд определяется внутренней структурой микропроцессора или интегрального блока, включающего систему микрокоманд. Существуют МП с фиксированным и изменяемым набором команд. По типу команд архитектуру процессоров можно разделить на RISC и CISC.

- **RISC — Reduced (Restricted) Instruction Set Computer** — процессоры (компьютеры) с сокращенной системой команд. Набор большого числа однородных регистров универсального назначения. Система команд относительно проста, коды инструкций имеют четкую структуру, как правило, с фиксированной длиной. Аппаратная реализация проста, легко конвейеризируется и разгоняется.
- **CISC — Complete Instruction Set Computer** — процессоры (компьютеры) с полным (сложным) набором инструкций. Регистры существенно неоднородны, широкий набор и разная длина команд усложняет декодирование инструкций, на что расходуются аппаратные ресурсы. Возрастает число тактов, необходимое для выполнения инструкций.

Рассмотрим обобщенную структуру микропроцессора:

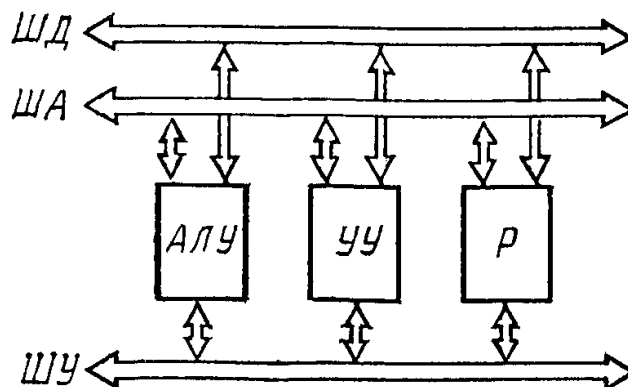


Рис. 1. Обобщенная структурная схема микропроцессора

Арифметико-логическое устройство (АЛУ) реализует арифметические (сложение, вычитание) и логические (сравнение, И, ИЛИ) операции над двумя числами и выдает результат операций.

Регистры (Р) или *блок регистров* предназначены для хранения и выдачи команд (*регистр команд*), адресов (*регистр адресов*) и данных (*аккумулятор*).

Устройство управления (УУ) служит для преобразования команд, поступающих из регистров и внешнего *запоминающего устройства (ЗУ)*, в сигналы, непосредственно воздействующие на все элементы МП и стимулирующие выполнение команд. Все блоки МП связаны между собой и с внешними устройствами тремя шинами: *шиной данных (ШД)*, *шиной адресов (ША)* и *шиной управления (ШУ)*. Шина состоит из набора параллельных проводников, к которым подключаются блоки МП и внешние устройства. Она служит для обмена информацией между этими блоками. Устройство управления определяет разделение во времени связей между блоками по одним и тем же проводникам (мультиплексирование).

Шина данных служит для обмена *операндами* — исходными элементами данных, под которыми можно понимать числовые данные или команды (также представляемые числами).

Шина адреса служит для передачи адресов — указаний (в виде чисел) местоположения ячейки памяти в запоминающем устройстве.

Шина управления служит для обмена сигналами управления.

Микропроцессорные комплекты оперируют информацией, представленной в двоичном коде в виде электрических сигналов. По количеству одновременно обрабатываемых двоичных сигналов определяется *разрядность микропроцессора* (по сути совпадает с *разрядностью шины данных*). Большинство распространенных МП рассчитаны на работу с 4, 8, 16 и 32-разрядными «словами».

Разрядность шины адреса определяет количество ячеек внешнего запоминающего устройства или число внешних устройств ввода-вывода к которым МП способен произвести непосредственное обращение (2^n).

Выполнение отдельных команд, в зависимости от типа процессора, реализуется одной или несколькими простейшими операциями — *микрокомандами*.

Микрокоманды – элементарные команды, определяющие выполнение микроопераций. Для каждой команды (CISC) можно составить микропрограмму, состоящую из последовательности микрокоманд.

Принцип работы синхронных устройств, по которому принято строить большие цифровые системы, подразумевает, что начало выполнения каждой микрооперации однозначно фиксируется во времени (синхронизируется) поступлением тактового (синхронизирующего) сигнала. Период поступления синхронизирующих импульсов (СИ) является, таким образом, минимальным временем между выполнением в системе двух последовательных микроопераций и служит единицей машинного времени - **тактом**.

Синхронные системы сложнее и, в принципе, медленнее асинхронных, но позволяют организовать четкую безошибочную работу больших цифровых систем, что практически невыполнимо в асинхронных системах из-за непредсказуемого разброса времени выполнения микроопераций, вызванного технологическим и др. разбросом параметров логических элементов.

Для того чтобы получить функционально законченное устройство, к микропроцессору необходимо добавить, как минимум, *устройство ввода — вывода информации (УВВ), запоминающее устройство (ЗУ) и тактовый генератор (ТГ)* (рис. 2).

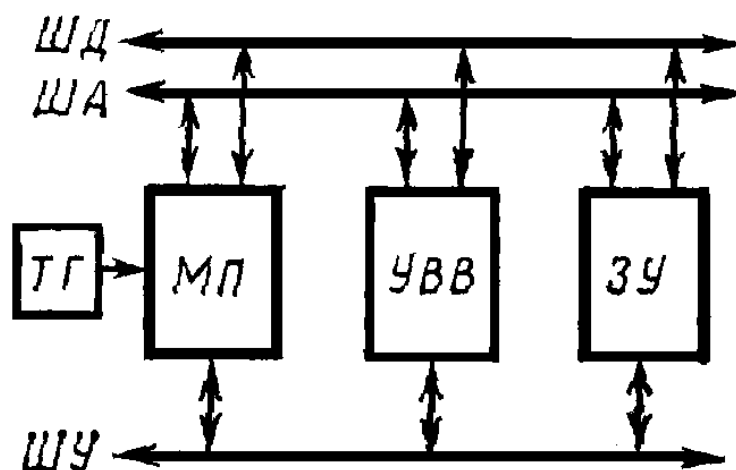


Рис. 2. Обобщенная структурная схема микро-ЭВМ

Систему изображенную на рис. 2 называют *микро-ЭВМ* или *микропроцессорной системой (МПС)*.

Тактовый генератор (*синхрогенератор*) синхронизирует через МП работу всех блоков системы.

Устройство ввода — вывода информации служит для ввода информации от внешних устройств в процессор или в ЗУ и для вывода информации во внешние устройства.

Запоминающее устройство — для приема, хранения и выдачи программы и данных. По характеру (способу) хранения различают оперативное ЗУ (ОЗУ) - энергозависимое и **постоянное ЗУ** (ПЗУ) – энергонезависимое.

Программа, по которой работает МП, обычно хранится в постоянном запоминающем устройстве (ПЗУ), в которое она заносится однократно при изготовлении ПЗУ. *Возможно применение репрограммируемых ПЗУ (РПЗУ), допускающих многократную запись и стирание информации (например, микросхемы К573РФ1 — К573РФ5), в которых стирание происходит при ультрафиолетовом облучении.* Для хранения утилит и данных применяют ОЗУ, в которые информация может быть введена микропроцессором или внешними устройствами (например, клавиатурой ввода, каналами связи с внешними объектами).

Устройства ввода — вывода представляют собой специализированные ИС высокой степени интеграции, выполняющие специфические функции обмена сигналами с внешними устройствами и позволяющие управлять своей работой программным путем. К ним относят параллельный и последовательный адаптеры, контроллеры клавиатуры, дисплея, диска, прямого доступа к памяти и т.д.

Указанные устройства обеспечивают обмен с широким кругом внешних устройств: различными накопителями, принтерами, модемами графопостроителями, дисплеями и т.п.

Математическое обеспечение микропроцессорного устройства — совокупность управляющих программ для эффективного использования системы. Набор управляющих программ обычно заносится в ПЗУ и при старте микро-ЭВМ осуществляет настройку компонентов системы и все операции начальной инициализации системы.

МПС являются многофункциональными системами, способными выполнять функции практически любого устройства с «жесткой» логикой.

Память в микропроцессорной системе

(ЗАПОМИНАЮЩИЕ УСТРОЙСТВА)

Для хранения информации в микропроцессорной системе применяют **запоминающие устройства или память**. Память можно разделить на два типа - внутреннюю и внешнюю.

Внешней называют память на магнитных и оптических дисках, магнитных лентах, перфолентах и перфокартах. Последние два типа в настоящее время из употребления практически вышли.

Внутренняя память компьютера обычно представлена в виде отдельных интегральных микросхем (ИМС) собственно памяти и элементов, включенных в состав других ИМС, не выполняющих непосредственно функцию хранения программ и данных - это и внутренняя память центрального процессора, и видеопамять, и регистры контроллеров различных устройств.

Внутреннюю или основную память можно также разделить на: **оперативное запоминающее устройство (ОЗУ)** или запоминающее устройство (ЗУ) с произвольной выборкой (ЗУПВ) и **постоянное ЗУ (ПЗУ)**. ОЗУ, кроме того, обозначается - (RAM, Random Access Memory), а ПЗУ - (ROM, Read Only Memory). Получила также распространение Флэш (Flash) память, имеющая особенности и ОЗУ и ПЗУ и энергонезависимая память (Nonvolatile - NV). Последнее название условно, так как ПЗУ и Флэш-память, также энергонезависимы. В ОЗУ коды в соответствии с решаемыми задачами постоянно изменяются и полностью пропадают при выключении питания. В ПЗУ хранятся управляющие работой ЭВМ стандартные программы, константы, таблицы символов и другая информация, которая сохраняется и при выключении компьютера.

ОЗУ подразделяются на статическую память (SRAM), динамическую (DRAM), регистровую (RG).

ПЗУ могут быть: масочными - запрограммированными на заводе изготовителе (ROM), однократно программируемыми пользователем ППЗУ (PROM или OTP), многократно программируемыми (репрограммируемыми) пользователем РПЗУ с ультрафиолетовым стиранием (EPROM) или с электрическим стиранием (EEPROM, Flash).

Для создания элементов запоминающих устройств, в основном, применяют большие интегральные схемы (БИС) со структурой МДП (металл-диэлектрик-полупроводник) на основе кремния (в связи с тем, что в качестве диэлектрика чаще всего используют его оксид SiO₂, то их обычно называют МОП (металл-оксид-полупроводник) структурами).

В зависимости от типа ЗУ элементом памяти (ЭП) может быть: **триггер, миниатюрный полупроводниковый конденсатор, транзистор с "плавающим затвором", плавкая перемычка** (или ее отсутствие). Упорядоченный набор ЭП образует ячейку памяти (ЯП). Количество элементов памяти в ячейке (длина слова) обычно кратно 2ⁿ (1, 4, 8, 16, 32, 64..), причем величины свыше 8-ми достигаются, обычно, группировкой микросхем с меньшим количеством ЭП. Количество ЭП в ЯП иногда называется длиной слова. Основными характеристиками микросхем памяти являются:

информационная емкость, быстродействие, энергопотребление.

Емкость ЗУ чаще всего выражается в единицах кратных числу 2¹⁰ = 1024 = 1К. Для длины слова равной биту (одному двоичному разряду) или байту

(набору из восьми бит) эта единица называется килобит или килобайт и обозначается Kb или KB.

При разрядности “ n ” шины адреса микропроцессорной системы непосредственно доступны 2^n ячеек памяти. Каждой из двух в степени “ n ” ячеек памяти однозначно соответствует “ n ”- разрядное двоичное число, называемое адресом ЯП.

Например, адресом 511-ой ячейки будет число $0001.1111.1111(BIN) = 511(DEC) = 1FF(HEX)$. В программах адреса употребляются в 16-ном формате. Емкость ЗУ также часто выражается произведением двух чисел $2^n * m$, где 2^n - число ячеек памяти, а m - длина слова ячейки, например $8K * 8$ (м/с 537PY17), т.е. 8192 ячеек размером в один байт. В некоторых справочниках для этой же микросхемы приводится обозначение емкости одной цифрой 64Кбит, что никак не отражает внутреннюю организацию этой микросхемы, такую же емкость могут иметь микросхемы с организацией $16K * 4$, $64K * 1$.

ОЗУ СТАТИЧЕСКОГО ТИПА

В качестве элемента памяти в ОЗУ статического используется простейший D-триггер или триггер типа защелка. Устройство триггера в интегральном исполнении по технологии МОП представлено на рисунке 1.

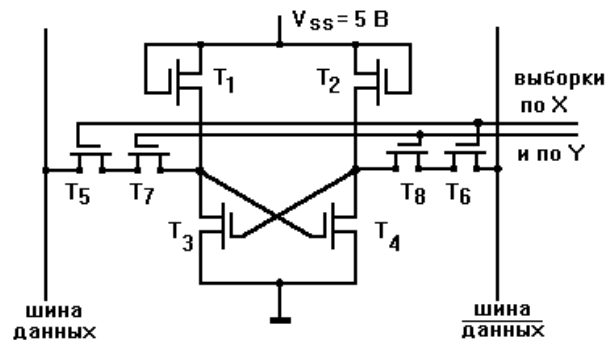


Рис.1 Ячейка статического ОЗУ.

Собственно триггер выполнен на транзисторах T1-T4 с ключами T5-T8 для доступа к шине данных. Причем T1-T2 - это нагрузки, а T3-T4 - нормально закрытые элементы. Сопротивление элементов T1-T2 легко регулируется в процессе изготовления транзистора путем подгонки порогового напряжения при легировании поликремниевого затвора методом ионной имплантации.

Рассмотрим в качестве примера статического ЗУ микросхему памяти типа K537PY10. В микросхеме 537PY10 каждая ЯП состоит из восьми триггеров, и располагаются ячейки на кристалле в виде прямоугольной матрицы (см. Рис. 2).

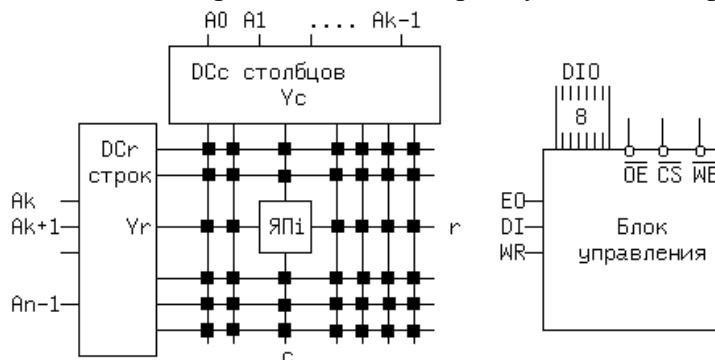


Рис. 2 Структура статического ОЗУ

На рисунке приняты обозначения: n -адресных входов ($A_0 \dots A_{n-1}$), DIO - двунаправленная восьмиразрядная шина данных, вход разрешения выходов - /OE (Output Enable, / - низкий активный уровень), чип вход выбора микросхемы - /CS (Chip Select) и вход разрешения записи - /WE (Write Enable), который часто обозначают по-другому - /WR / RD (Write / Read), подчеркивая этим, что при низком значении сигнала на этом входе производится запись байта, а при высоком уровне - чтение. EO, DI, WR - внутренние сигналы, вырабатываемые блоком управления чтением/записью/хранением.

Доступ к произвольной ЯПj двух обычных дешифраторов, причем k -адресных линий заводится на дешифратор столбцов (DCc), а оставшиеся $n-k$ линий подключены к дешифратору строк (DCr). Количество строк и столбцов будет соответственно равно 2^{n-k} и 2^k , т.е. общее количество, обслуживаемых ЯП, равно $2^k * 2^{n-k} = 2^n$.

На рисунке 3 показана схемотехника фрагмента внутренней структуры микросхемы. Здесь же дано условное графическое обозначение (УГО) микросхемы.

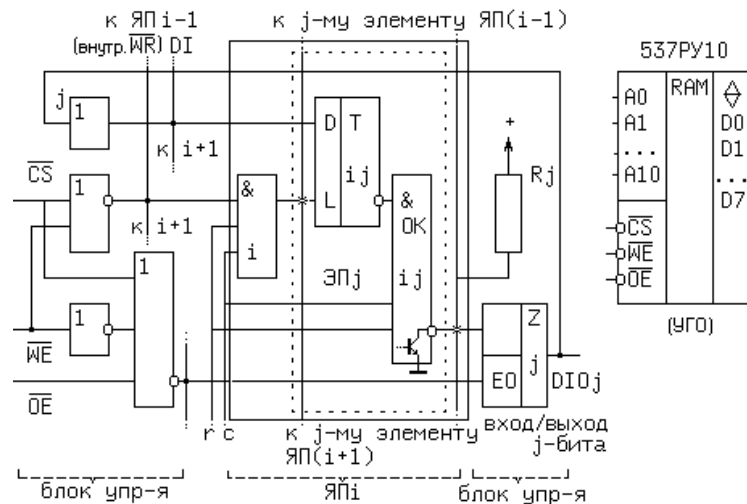


Рис. 3 Фрагмент внутренней структуры и УГО микросхемы.

На рисунке схемы с открытым коллектором и третьим состоянием обозначены ОК и Z - соответственно. Пунктиром выделен один (j -ый) из восьми элементов i -ой ячейки памяти. Схема И с номером $i = (r * 2^k + c)$ является одним из 2^n выходных узлов сдвоенного дешифратора, где r и c - номера строк и столбцов матрицы. Инверсный вход - /CS, во всех микросхемах, где он встречается, служит для приведения схемы в рабочее состояние низким уровнем сигнала на нем.

Если /CS = 1 (пассивный уровень), микросхема - не выбрана, и операции с ней производить невозможно. Из рисунка 3 видно, что в этом случае на L-входе D-триггера - ноль, запись невозможна и триггер хранит ранее записанный бит, прочесть выходной код - Q тоже нельзя, т.к. на прямом входе EO разрешения выхода запрещающий нулевой сигнал и вход/выход DIOi находится в третьем (Z) состоянии. С поступлением /CS = 0, схемы ИЛИ-НЕ разблокируются и далее логика работы зависит от значений сигналов /WE и /OE.

В режиме записи сигнал /WE = 0. Поэтому независимо от значения сигнала /OE на входе схемы, внутренний сигнал EO, тоже равен 0, и чтение данных во время записи невозможно. На верхнем входе элемента Иi - единица и, если на адресных входах код $A_{n-1}, A_{n-2}, \dots, A_1, A_0(\text{BIN}) = i(\text{DEC})$, то сигналы

на линиях Y_r и Y_c тоже равны 1 и триггер ij прозрачен для записи входной информации DIO_j .

В режиме чтения $/WE=1$, $/OE=0$ и при $Y_r=Y_c=1$, выходной сигнал $/Q$ после инверсии элементом $И_{ij}$ с открытым коллектором проходит на выход DIO_i . Следует обратить внимание на то, что выходы всех 2^n j -ых элементов памяти должны подключаться к общему j -му выводу микросхемы - DIO_j . Такое объединение выходов возможно с помощью схемного либо монтажного **И (ИЛИ)**. Монтажное И (ИЛИ) не требует дополнительных схем и может выполняться на элементах с открытым коллектором или с третьим состоянием. Внутри рассматриваемой схемы j -е выходы ЭП объединены на общем резисторе R_j , служащем нагрузкой элементов $И-НЕ_{ij}$ с открытым коллектором. Для увеличения информационной емкости, отдельные микросхемы группируются в банки и их одноименные выходы должны объединяться. **По этой причине выходы всех микросхем памяти также выполняются с открытым коллектором либо с третьим состоянием.**

ОЗУ ДИНАМИЧЕСКОГО ТИПА

В качестве элемента памяти используется микроконденсатор в интегральном исполнении (Рис. 4), размеры которого значительно меньше D-триггера статической памяти. При этом для хранения одного бита в ячейке **DRAM**, кроме накопительного конденсатора, необходимы всего 1-2 транзистора. По этой причине, **при одинаковых размерах кристалла, информационная емкость DRAM выше, чем у SRAM.**



Рис. 4 Запоминающая ячейка динамического ОЗУ.

Количество адресных входов и габариты должны увеличиться. Чтобы не допустить этого, адресные линии внутри микросхемы разбиваются на две группы, например старшая и младшая половина. Две одноименные k -линии каждой группы подключаются к двум выходам внутреннего k -го демультиплексора "1 в 2", а его вход соединяется с k -ым адресным входом микросхемы. Количество адресных входов, при этом уменьшается в два раза, но зато передача адреса в микросхему должна производиться,

- во-первых, в два приема, что несколько уменьшает быстродействие,
- во-вторых, требуется дополнительный внешний мультиплексор адреса.

В процессе хранения бита конденсатор разряжается. Чтобы этого не допустить заряд необходимо поддерживать (регенерировать). Обычно для этого периодически опрашивают все ячейки DRAM за время, специфическое для каждой отдельной БИС.

Суммируя, можно перечислить, чем отличается динамическое ОЗУ от статического:

- 1) мультиплексированием адресных входов,
- 2) необходимостью регенерации хранимой информации,
- 3) повышенной емкостью (до нескольких Мбит),
- 4) более сложной схемой управления.

На рисунке 5 приведена структурная схема, а на рисунке 6 условное обозначение микросхемы 565РУ7 емкостью 256К*1 (2^{18} К) и способ подключения 18-ти линий адреса к девяти адресным входам с помощью 9-ти мультиплексоров "2 в 1", например трех счетверенных селекторов-мультиплексоров типа 1533КП16.

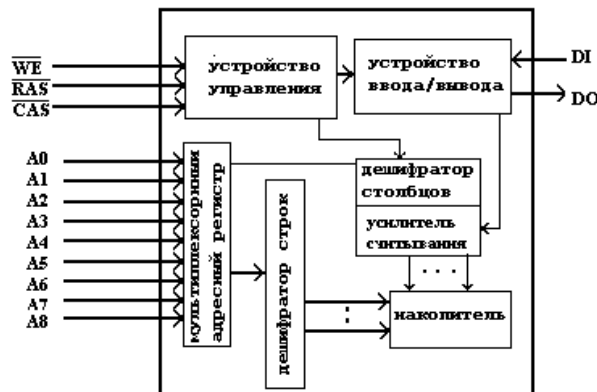


Рис. 5. Структурная схема микросхемы динамического ОЗУ.

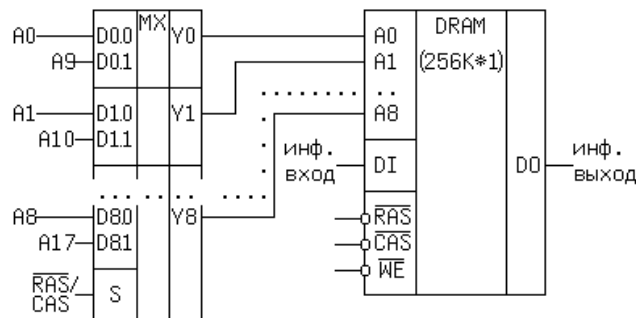


Рис. 6. Способ подключения микросхемы динамического ОЗУ.

Элементы памяти расположены на кристалле в виде матрицы $512 * 512 = 2^9 * 2^9$, управляемой двумя линейными дешифраторами строк и столбцов, каждый с 9-ю адресными входами. Если сигнал строка/столбец /RAS / CAS на входе выбора S мультиплексора, равен нулю, то $A(0..8) = Y(0..8)$ и в микросхему передается адрес строки. Этот адрес фиксируется отрицательным фронтом строка адреса строк /RAS. При /RAS / CAS = 1 на выходы мультиплексора передается адрес столбцов $A(9..17)$, который защелкивается отрицательным перепадом строка адреса столбцов /CAS. Вход /WE управляет записью/чтением. Оперативная память персональных ЭВМ - (SIMM, EDO, SDRAM..) является динамической памятью. Время обращения к ней меньше 10 нс, а емкость достигает 256М в одном корпусе.

РЕПРОГРАММИРУЕМОЕ ПЗУ

Микросхемы РПЗУ допускают **многократное** (до сотен тысяч циклов) **перепрограммирование**. Это свойство обеспечивается применением ЭП на МОП транзисторах с "плавающим затвором" (Рис. 6).

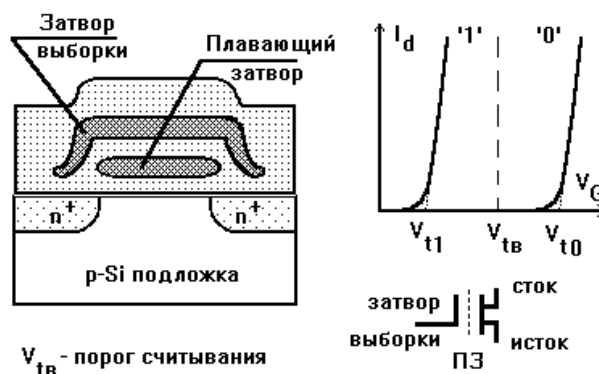


Рис. 7. МОП транзистор с плавающим затвором.

Это р-канальный нормально закрытый МОП прибор. Здесь же показаны вольт-амперные характеристики (ВАХ) транзистора в состоянии логических единицы и нуля (до и после записи информационного заряда). Плавающий затвор представляет собой область поликремния, окруженную со всех сторон диэлектриком, т.е. он электрически не связан с другими электродами и его потенциал "плавает". Обычно толщина нижнего диэлектрического слоя составляет от десятков до сотен ангстрем. Это позволяет в сильном электрическом поле инжектировать электроны в плавающий затвор:

- или сквозь потенциальный барьер Si-SiO₂ путем квантово-механического туннелирования;
- или над барьером "горячих" носителей, разогретых в поперечном или продольном поле при обратимом пробое кремниевой подложки.

Положительное смещение на верхнем затворе (относительно полупроводниковой подложки) вызовет накопление электронов в плавающем затворе при условии, что утечка электронов через верхний диэлектрический слой мала. Лавинный пробой подложки вблизи стока может приводить к неоднородной деградации транзистора и, как следствие, к ограничению по числу переключений элемента памяти. МДП-транзистор с плавающим затвором может быть использован в качестве элемента памяти с временем хранения, равным времени диэлектрической релаксации структуры, которое может быть очень велико и, в основном, определяется низкими токами утечки через барьер Si-SiO₂. Записанная информация сохраняется в течение нескольких (до 10-12) лет. Такой элемент памяти обеспечивает возможность непрерывного считывания без разрушения информации, причем запись и считывание могут быть выполнены в очень короткое время.

Информация считается стертой, если на выходах всех ЭП высокий уровень сигнала. В режиме программирования, на выбранный по адресной шине ЭП, куда необходимо записать ноль, подается импульс. Стирание информации (возврат структуры в исходное состояние) может осуществляться:

- ультрафиолетовым излучением с энергией квантов более 5.1 эВ (ширина запрещенной зоны нитрида кремния) через кварцевое окно (EPROM),
- либо электрически, подачей на структуру импульса напряжения, противоположного по знаку записываемому (EEPROM).

При этом все ячейки переводятся в состояние "1".

В соответствии с ГОСТом такие ИМС имеют в своем названии литеры РФ и РР соответственно. Основными факторами, влияющими на запись и хранение заряда, являются электрическое поле, температура и радиация. Количество электрических циклов "запись-стирание" обычно не менее 10^5 .

Одной из микросхем этого типа является EPROM 573РФ2 с организацией (2К * 8) и тристабильными выходами.

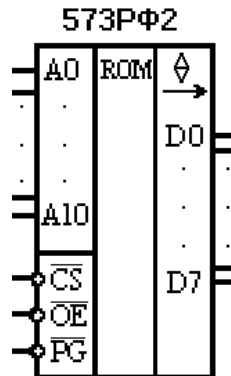


Рис. 8. Микросхема ПЗУ типа 573РФ2

Вывод /PG микросхемы используется при программировании её с помощью специальных устройств (программаторов) и при включении в микропроцессорную систему обычно не используется (на вывод подается уровень логической "1").

В **Flash**-памяти толщина изоляции "плавающего затвора" менее 100 ангстрем, поэтому при перепрограммировании используется туннельный эффект и запись информации в микросхему может осуществляться непосредственно в составе микропроцессорной системы.

ОДНОКРАТНО ПРОГРАММИРУЕМЫЕ ПЗУ ППЗУ (PROM)

ПЗУ такого типа (Рис. 9) в качестве элементов памяти имеют набор плавких перемычек, которые в процессе программирования пережигаются импульсами тока.

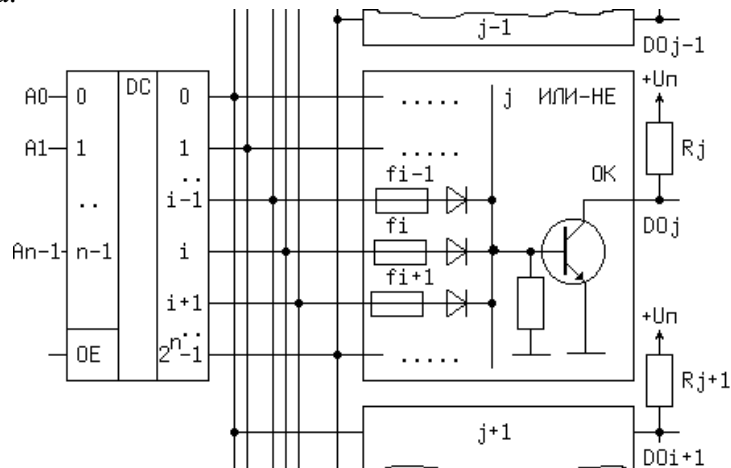


Рис.9. Схема однократно программируемого ПЗУ.

Для любого значения адресных сигналов найдется единственный выход дешифратора "i" на котором сигнал $Y_i=1$, на остальных выходах будут нули. Потенциал базы j-транзистора будет зависеть в этом случае только от наличия или отсутствия переключки f_i . Если переключка есть ($f_i=1$), то на базе высокий уровень сигнала, транзистор открыт и выходной сигнал $DO_j = 0$. Если переключки нет ($f_i=0$), то $DO_j=1$. Пережиганием переключек в соответствующих j-битах всех адресов, в микросхему записывается программа и (или) данные.

Учитывая внутреннее строение ППЗУ и тот факт, что транзистор в схеме с общим эмиттером представляет собой инвертор, несложным путем можно доказать, что **с помощью ПЗУ с n-адресными входами и m-выходами можно реализовать любые m-логических функций с n-переменными** (учитывая инверсию сигнала выходным каскадом). Особенно удобно реализовывать дешифраторы для семисегментных и сложных матричных индикаторов, а также дешифраторы, реализующие сложные функции в контроллерах видео-ОЗУ.

ЭНЕРГОНЕЗАВИСИМАЯ ПАМЯТЬ (NVSRAM)

Всякая память, сохраняющая данные при отключении внешнего источника питания, может считаться энергонезависимой - NonVolatile Memory, однако этот термин больше утвердился за статической оперативной памятью:

- с встроенной в микросхему литиевой батареей большой емкости (Рис. 10);
- с дополнительной EEPROM на том же кристалле, причем обмен данными между SRAM и EEPROM производится либо программно, либо автоматически при падении/восстановлении напряжения питания.

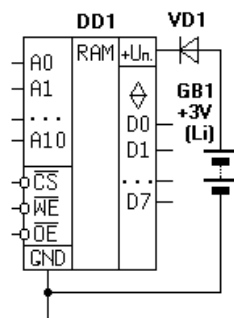


Рис. 10. NVSRAM с элементом питания.

На рисунке 10 изображен вариант энергонезависимой памяти, применяемой для сохранения настроек BIOS в персональных компьютерах. Диод VD1 выполняет роль безинерционного переключателя: при выключении питания диод открывается и подсоединяет к выводу питания статического CMOS ОЗУ DD1 литиевую батарею, обеспечивающую режим хранения данных; при включении питания диод запирается напряжением питания $+U_{\Pi} = 5 \text{ В}$, превышающим напряжение +3 В батареи, вследствие чего последняя оказывается отключенной от цепи питания ОЗУ. Емкости литиевой батареи обычно хватает практически на весь срок эксплуатации материнской платы ЭВМ, с установленной на ней микросхемой CMOS.

УВЕЛИЧЕНИЕ РАЗРЯДНОСТИ ЯЧЕЙКИ ПАМЯТИ (СЛОВА)

Если требуется хранить данные размером в n-бит, а длина слова ячейки памяти m-бит ($n > m$), то прибегают к наращиванию длины слова. Делается это путем

объединения **n/m** микросхем в группы, причем все одноименные входы, кроме информационных, соединяются между собой. Например, если требуется динамическая память емкостью 256К с длиной слова равной байту, то необходимо объединить $8 / 1 = 8$ микросхем типа 565РУ7.

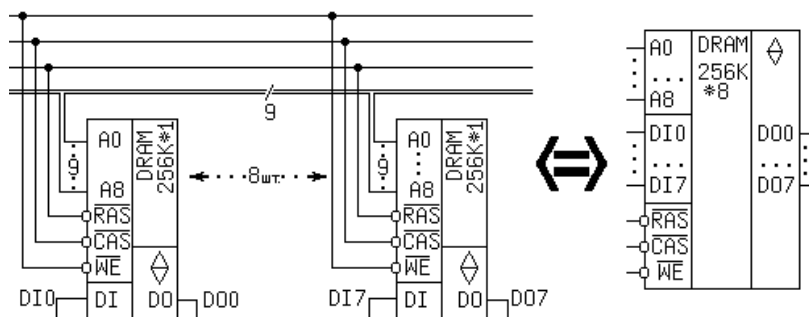


Рис. 11. Пример наращивания длины слова.

На рисунке девять линий адреса показаны в виде **шины - т.е. группы проводников, объединенных по функциональному признаку**. Входы DI0 – DI7 и выходы DO0-DO7 могут быть объединены в единую шину адреса, поскольку выходы обладают Z-состоянием, а управление чтением/записью управляется через единый вход /WE, что не позволяет микросхемам одновременно выдавать данные на шину и в то же время считывать с нее данные.

Микросхемы памяти в составе микропроцессорной системы

Микропроцессор КР580ВМ80 (i8080) рассматривает адресное пространство в виде непрерывной, замкнутой в кольцо, последовательности ячеек памяти. 16 линий адреса позволяют непосредственно адресовать $2^{16} = 65536$ ячеек (0000H – 0FFFFH). После ячейки 0FFFFH должна следовать ячейка 10000H, но 16 разрядная шина адреса отбрасывает 17-й разряд, и переход происходит по адресу 0000H. Адресное пространство микропроцессора удобно рассматривать в виде карты памяти, укладывая в ней ячейки памяти по 16 (Рис.12). Такое представление памяти вызвано лишь удобством представления и не имеет под собой никакой физической интерпретации.

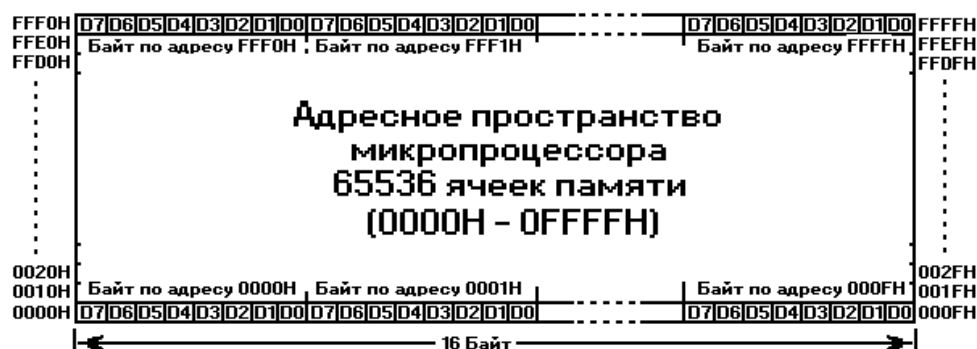
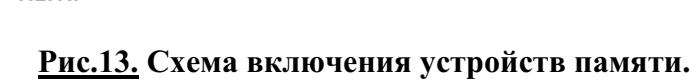


Рис.12. Адресное пространство микропроцессора.



Микросхема-дешифратор К555ИД7 типа код-позиция, дешифруя стар-

Поскольку микропроцессор КР580ВМ80 (і8080) по включению питания

На рисунке 14 представлена карта памяти микропроцессорной системы, иллюстрирующая рассмотренную выше схему включения устройств памяти.

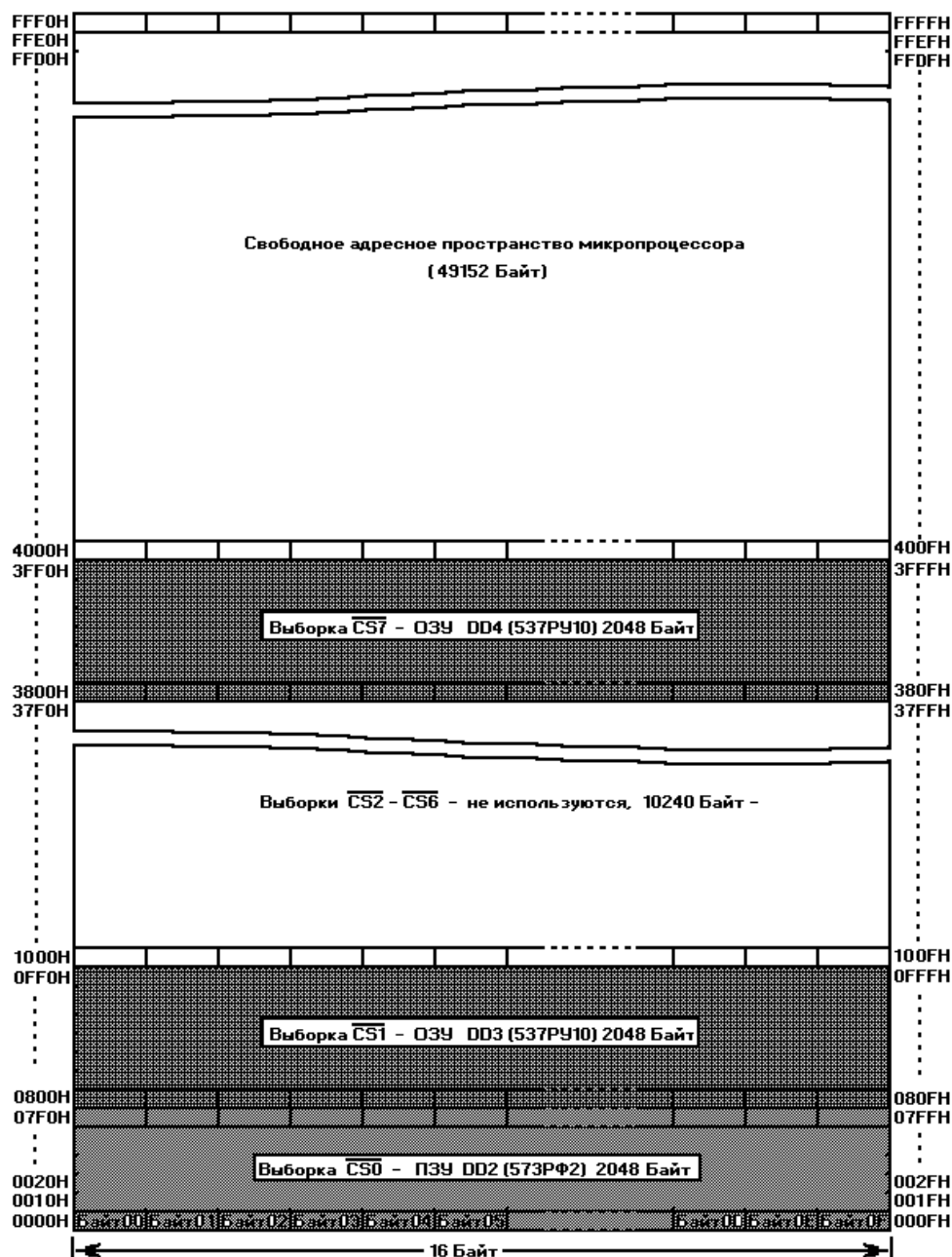


Рис. 14. Карта памяти с подключенными ЗУ.

Как было упомянуто ранее - с помощью ЗУ можно реализовать любые логические функции, в том числе и дешифраторы. В этой связи представляется интересной замена дешифратора DD1 в схеме на рисунке 13 микросхемой ОЗУ, запрограммированной на эмуляцию его функций. В таком случае возможно оперативно менять карту памяти микропроцессорного устройства, прошивая в ОЗУ, эмулирующее дешифратор, таблицы данных, согласно которым микросхемы ЗУ будут занимать иные места в адресном пространстве микропроцессорной системы. Если учесть, что устройства ввода/вывода (или периферийные адаптеры) доступны микропроцессору как ячейки ЗУ, то применение ОЗУ в качестве дешифратора внешних устройств позволяет создать микропроцессорную систему с гибкой картой памяти. О преимуществах такого способа дешифрации, устройстве и подключении периферийных адаптеров будет рассказано в следующих лекциях.

МИКРОПРОЦЕССОР KP580BM80 (i8080)

Микропроцессорная БИС KP580BM80 представляет собой однокристалльный 8-разрядный микропроцессор (МП) с двумя магистралями: однонаправленной 16-разрядной адресной магистралью (МА), двунаправленной 8-разрядной магистралью данных (МД), и 12 сигналами управления (шесть входных и шесть выходных). Канал адреса обеспечивает прямую адресацию внешней памяти объемом до 65536 байт, 256 устройств ввода и 256 устройств вывода. Микропроцессорная БИС рассчитана на выполнение логических и арифметических операций с 8-разрядными числами в двоичной и десятичной системах счисления, а также операций с двойной разрядностью (с 16-разрядными числами). Максимальная частота тактирования микропроцессора достигает 3,5 МГц.

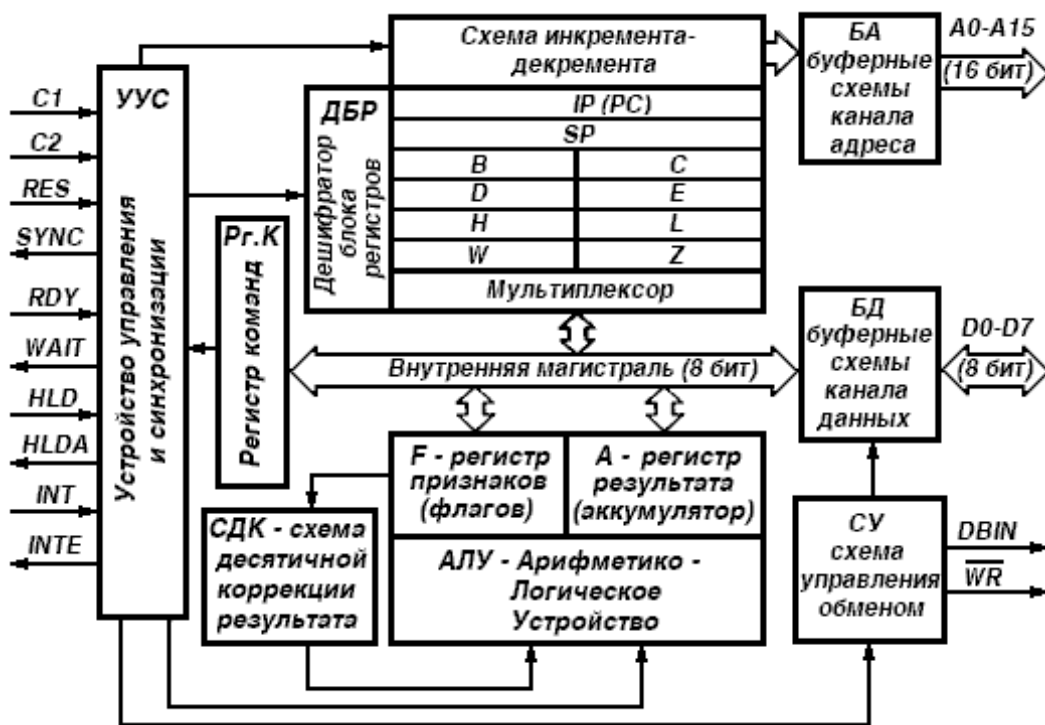


Рис. 1.

Структура МП БИС KP580ИК80 приведена на рис. 1. Микропроцессорная БИС состоит из следующих функциональных блоков (рис. 1):

- 1) регистров и логических цепей их выбора;
- 2) арифметически-логического блока;
- 3) регистра команд и устройства управления (УУ);
- 4) буферов МД (БФД) и МА (БФА).

Секция регистров включает в себя шесть 16-разрядных регистров: *W, Z, B, C, D, E, H, L, PC* (программный счетчик) и *SP* (указатель стека). Шесть 8-разрядных регистров общего назначения составлены в пары регистров: *H, L; D, E; B, C*. Они могут быть использованы в программах как отдельные 8-разрядные регистры, а также как три 16-разрядных регистра (*H; D; B*).

Программный счетчик *PC (IP)* содержит текущий адрес памяти, к которому обращается программа. Содержание *PC (IP)* автоматически изменяется в течение каждого цикла команды.

Указатель стека *SP* содержит адрес памяти, начиная с которого ее можно применять для хранения и восстановления содержания программно-доступных регистров МП БИС.

Арифметически-логический блок (АЛБ) выполняет арифметические и логические операции под воздействием устройства управления МП БИС. Он включает в себя 8-разрядное *АЛУ*, схему десятичной коррекции *ДК*, построенную на базе ПЗУ, 5-разрядный регистр признаков, регистр-аккумулятор *А*, буфер аккумулятора *БФА* и буферный регистр *БФР*. Арифметически-логический блок позволяет осуществить арифметические операции сложения, вычитания, а также основные логические операции (И, ИЛИ, исключающее ИЛИ) и сдвиг. При проведении операций одно число всегда берется из буфера аккумулятора, а другое — из буферного регистра. По результату выполнения арифметически-логических операций АЛБ устанавливает в регистре признаков пять разрядов.

Признак переноса (**Carry — C**) устанавливается в единицу, если при выполнении команд появляется единица переноса из старшего разряда.

Дополнительный признак переноса (**Auxiliary carry — AC**) устанавливается в единицу, если при выполнении команд возникает единица переноса из третьего разряда числа. Состояние разряда может быть проанализировано лишь командой десятичной коррекции числа.

Признак знака (**Sign — S**) в машинном слове можно представить числом от —128 до +127. В этом случае седьмой (старший) разряд числа — его знак. Единица в седьмом разряде при такой записи будет указывать на отрицательное число, а ноль — на положительное.

В разряд нулевого признака (**Zero — Z**) записывается единица, если при выполнении команды результат равен нулю.

В разряд признака четности (**Parity — P**) записывается единица, если при выполнении команды количество единиц в разрядах результата будет четным.

Регистр команд РзК и дешифратор команды **ДШК** используются в МП БИС для получения и дешифрации кода команды. При извлечении команды первый байт, содержащий ее код, помещается в регистр команды и поступает на дешифратор команд. Дешифратор совместно с устройством управления и синхросигналами Φ_1 и Φ_2 формирует управляющие сигналы для всех внутренних блоков МП БИС, а также его выходные сигналы управления и состояния.

Выполнение каждой команды производится МП БИС в строго определенной последовательности, определяемой кодом команды, и синхронизируется во времени сигналами Φ_1 и Φ_2 тактового генератора.

Период синхросигналов Φ_1 или Φ_2 называется **машинным тактом (T)**. Длительность машинного такта *T* может быть установлена произвольно в диапазоне от 0,3 до 2 мкс.

Функциональное назначение внешних выводов МП БИС КР580ИК80:

A0 — A15 — адресная магистраль (выходы с тремя состояниями), обеспечивающая адресацию к любой из 2^{16} 8-разрядной ячейки памяти или внешнего устройства (ВУ);

D0 — D7 (в отечественной литературе — **D0 — D7**) — двунаправленная магистраль данных, используемая для обмена информацией с памятью или ВУ;

SYNC (в отечественной литературе — **СИНХР** (синхронизация)) — на этом выходе МП БИС формируется сигнал синхронизации в начале каждого машинного цикла;

DBIN (Data Bus In или READ, в отечественной литературе — П(прием)) — сигнал активного уровня на этом выходе указывает на готовность МП БИС к приему данных, иначе — сигнал чтения;

WAIT (в отечественной литературе — **ОЖД**) (ожидание) — активный сигнал на этом выходе указывает, что МП находится в состоянии ожидания;

WR (WRITE, в отечественной литературе — Зп) — на этом выходе МП БИС сигнал активного уровня указывает, что данные выданы МП БИС и установлены на МД и могут быть записаны в ВУ, иначе — сигнал записи;

HLDA (Hold Acknowledgement, в отечественной литературе — П.ЗХ) (подтверждение захвата) — на этом выходе МП БИС активный сигнал появляется в ответ на сигнал **HOLD** (запрос захвата) и указывает, что ШД и ША находятся в состоянии высокого сопротивления;

INTE (Interrupt Enable, в отечественной литературе — Р.Пр) (разрешение прерывания) — активный сигнал на этом выходе указывает на состояние внутреннего триггера разрешения прерывания МП БИС. Состояние триггера может быть установлено программно с помощью команд **EI, DI**. При уровне «0» на этом выходе прием запросов прерывания МП БИС невозможен;

READY (в отечественной литературе — **Г**) (готов) — активный сигнал на этом входе информирует о готовности ВУ к обмену информацией с МП БИС. При уровне «0» МП БИС будет находиться в состоянии ОЖИДАНИЕ.

HOLD (в отечественной литературе – **3.3X**) (запрос захвата) — вход, используемый для подачи сигнала **3.3X** на переход МП БИС в состояние ЗАХВАТ, в котором МА и МД переходят в третье состояние (высокое сопротивление). Обычно состояние используется для организации обмена информацией по каналу прямого доступа к памяти;

INT (в отечественной литературе – **3.Пр**) (запрос прерывания) — вход, используемый для подачи сигнала запроса прерывания. Сигнал поступает от внешнего устройства на прерывание выполнения основной программы и переход на выполнение подпрограмм обслуживания прерывания. Сигнал запроса прерывания не воспринимается МП БИС при работе его в режимах ЗАХВАТ, ОЖИДАНИЕ или нулевом состоянии внутреннего триггера разрешения прерывания;

RES (Reset, в отечественной литературе – Сбр.) — вход, по которому поступает сигнал на начальную установку МП БИС, при этом обнуляются его программный счетчик, внутренние триггеры, формирующие сигналы **INTE** и **HLDA**;

C₁, C₂ (в отечественной литературе – **Φ₁** и **Φ₂**) — входы для подачи тактовых сигналов **Φ₁** и **Φ₂**. Последние являются не пересекающимися во времени сигналами, определяющими тактовую частоту работы МП БИС.

Входы источников питания **Ucc1, Ucc2, U₁₀, GND** МП БИС соответственно: +5В, +12В, —5В, нулевой потенциал (земля).

Режимы работы МП БИС КР580ИК80.

Начальная установка. В течение всего времени, пока на входе **RES** установлен уровень «1», все операции в МП БИС будут приостановлены. За это время обнуляются программный счетчик, триггер разрешения прерываний и внутренняя логика, связанная с формированием сигнала **П.3X**. Так как программный счетчик содержит к этому времени число 0000H, то первая выполняемая команда будет считана по адресу 0000H.

Режим работы ОЖИДАНИЕ. Если к моменту появления отрицательного фронта **Φ₂** на такте **T₂** сигнал на входе **ГОТОВ** будет иметь нулевой потенциал, то МП БИС на такте **T₃** переходит в режим ожидания. В этом режиме приостанавливается процесс обработки данных и формируется единичный сигнал на выходе **ОЖИДАНИЕ**, подтверждающий это состояние. Время ожидания выражается целым числом тактов и может длиться бесконечно долго. На каждом такте в этом режиме при отрицательном фронте **Φ₂** будет проверяться состояние сигнала на входе **ГОТОВ**.

Работа при выполнении команды HALT. Режим **ОСТАНОВ** является результатом выполнения команды **HALT**. Находясь в этом режиме, магистрали МП БИС не отключаются от магистралей микро-ЭВМ, а процессор просто отмечает время, в течение которого он ничего не выполняет. Состояние микро-ЭВМ в этом режиме аналогично состоянию **ОЖИДАНИЕ**, за исключением того, что это состояние устанавливается программно. Выход из режима останов осуществляется по сигналу сброс или по сигналам прерывания.

Обслуживание запросов на прерывание. Внешние устройства выдают сигнал запроса прерывания асинхронно по отношению к работе МП БИС путем подачи уровня «1» на вход **3.Пр**. Запрос может быть удовлетворен только в том случае, если МП БИС находится в состоянии разрешения восприятия запросов прерывания. МП БИС автоматически устанавливается в состояние запрета восприятия запросов прерывания после выполнения начальной установки, а также после начала обслуживания запроса прерывания. Управление состоянием восприятия запросов прерывания может осуществляться программно с помощью команд: **EI** — «Разрешение прерываний» и **DI** — «Запрет прерываний».

МП БИС указывает на возможность восприятия прерываний на выходе **Р.Пр**. Если на этом выходе присутствует уровень «1», то запросы прерывания могут быть восприняты, в противном случае (уровень «0» на выходе **Р.Пр**) запросы восприниматься не будут.

Если запрос прерывания воспринят, то начинается цикл обслуживания прерывания. На первом машинном цикле обслуживания прерывания МП БИС получает по МД микро-ЭВМ код прерывания. Имеется два способа задания кода прерывания: с помощью команды **RST<N>** и с помощью команды **CALL<A><A₂>** (векторный способ).

Получив команду на первом машинном цикле, МП БИС записывает в область памяти, отведенной под стек, текущее значение программного счетчика **РС**. Сохранение содержания **РС** в стеке необходимо для возврата к прерванной программе после окончания выполнения подпрограммы обслуживания прерывания. Возврат можно осуществить

с помощью команды **RET** (RETURN), которая загружает содержание двух верхних ячеек стека в **PC**. Таким образом, каждая подпрограмма обслуживания прерываний должна в конце иметь команду **RET**.

Работа в режиме ЗАХВАТ. Режим ЗАХВАТ используется для останова управления МП БИС работой микро-ЭВМ. При работе в этом режиме выходные магистрали МП БИС переводятся в третье состояние и отключаются от магистралей микро-ЭВМ, обеспечивая тем самым возможность обмена информацией с микро-ЭВМ по каналу прямого доступа к памяти.

Режим ЗАХВАТ инициируется подачей сигнала «1» на вход управления $\bar{3}$. ЗХ МП БИС подтверждает переход в режим ЗАХВАТ путем выдачи сигнала «1» на выходе Π . ЗХ.

Состояние ЗАХВАТ будет длиться целое число тактов. Установка низкого уровня потенциала на входе $\bar{3}$.ЗХ проверяется на каждом сигнале Φ_2 , и при его наличии на следующем импульсе режим захвата будет прекращен и МП БИС перейдет к выполнению очередного машинного цикла.

Благодаря своей удачной конструкции, микропроцессор K580 и по сей день успешно используется в микроконтроллерах и различных системах управления процессами.

Типовая схема включения МП K580BM80 приведена на рисунке 2.

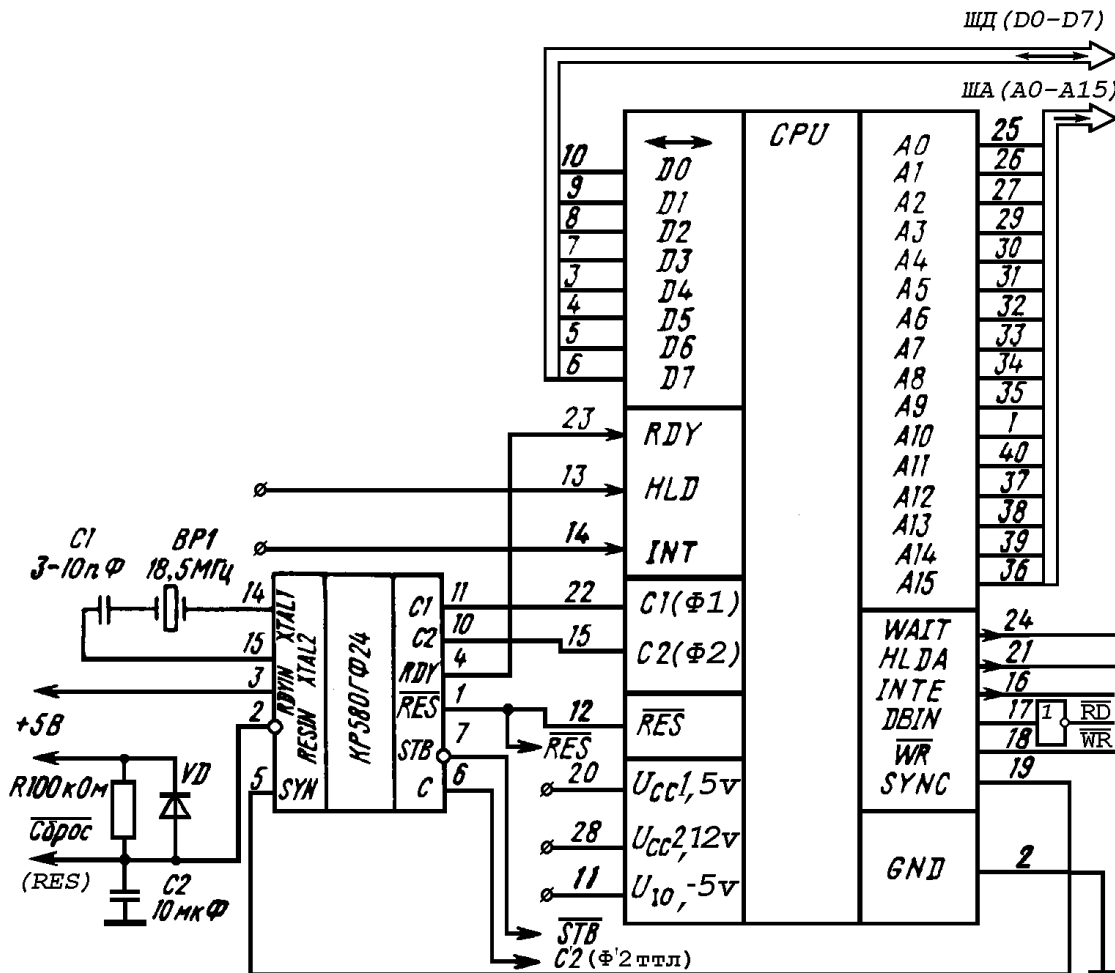


Рис. 2. Схема включения микропроцессора K580BM80.

Организация магистралей микро-ЭВМ. Обмен информацией между МП БИС и ВУ организуется с помощью трех магистралей микро-ЭВМ: МА, МД и МУ. Ввиду малой нагрузочной способности, а также ограниченного числа выводов МП БИС в микро-ЭВМ необходимо использовать дополнительные схемы для организации этих магистралей (системный контроллер КР580ВК28 и тристабильные шинные формирователи КР580ВА86). При построении простой системы и при условии, что нагрузка на 1 вывод МП не превышает 1 вход ТТЛ, указанные схемы необязательны.

При отладке устройств, построенных на МП БИС КР580ВМ80, может понадобиться пошаговое прохождение программы с анализом сигналов на шине данных и шине адреса. Такую возможность предоставляет устройство, схема которого изображена на рис. 3.

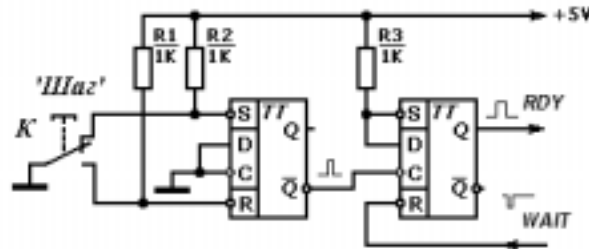


Рис. 3. Схема устройства для пошаговой отладки

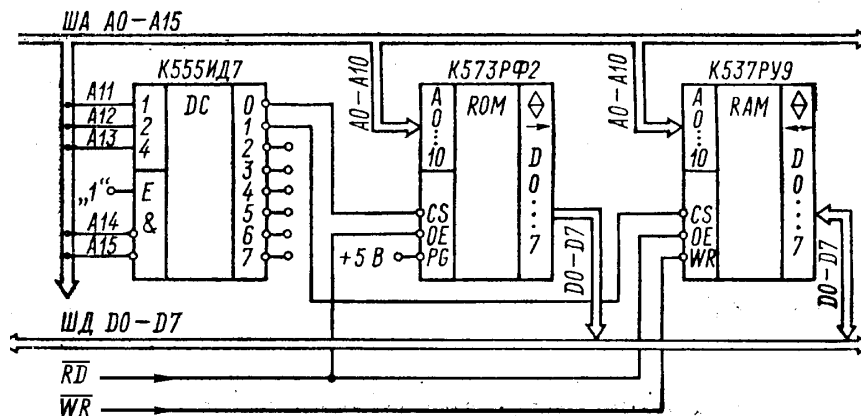
Будучи подключенным к выходу МП **WAIT** и ко входу **RDYIN** синхрогенератора КР580ГФ24 (рис. 2.) устройство работает следующим образом: при нажатии на кнопку “Шаг”, формируется сигнал **RDY** для МП и последний выполняет действия в цикле. Низкий уровень по выходу **WAIT** сбрасывает активный уровень сигнала **RDY** и МП начинает выполнять такты ожидания, при этом на шине данных и на шине адреса удерживаются действительные значения данных и адреса, что позволяет проконтролировать их с помощью измерительных приборов.

Магистраль адресов является однонаправленной магистралью, выдача адресов на которую осуществляется лишь МП БИС или схемой прямого доступа к памяти при работе микро-ЭВМ с каналом прямого доступа к памяти. Малая нагрузочная способность МА требует применения буфера.

По **магистрале данных** (МД) организуется двунаправленный обмен данными между МП БИС и памятью или ВУ. Максимальная нагрузочная способность МД невелика, что не позволяет непосредственно подключать к ней системы с большой емкостью памяти и широким набором периферийных устройств. Для увеличения нагрузочной способности МД микро-ЭВМ необходимо также применять двунаправленные буферные формирователи.

Магистраль управления состоит из пяти шин, сигналы на которых формируются из записанного слова состояния МП БИС и его выходных сигналов **RD** и **WR**.

Как обсуждалось ранее, для построения микропроцессорной системы необходимо подключение к МП запоминающих устройств (ПЗУ и ОЗУ), а также устройств ввода-вывода. Схема подключения микросхем памяти приведена на рис. 4.



О дешифрации адресного пространства микро-ЭВМ было подробно рассказано в лекции, посвященной запоминающим устройствам, в этой связи данная схема принципиальных особенностей не имеет.

Устройства ввода-вывода (УВВ) микропроцессорной системы могут быть адресованы как ячейки памяти в том случае, если в составе МПС отсутствует БИС системного контроллера КР580ВК28, осуществляющая разделение пространств памяти и пространства УВВ. Подключение УВВ в пространство ячеек памяти иллюстрирует рисунок 5. Заметим, что функции простейших устройствами ввода вывода могут выполнять обычные микросхемы регистров, соответствующим образом включенные в МПС.

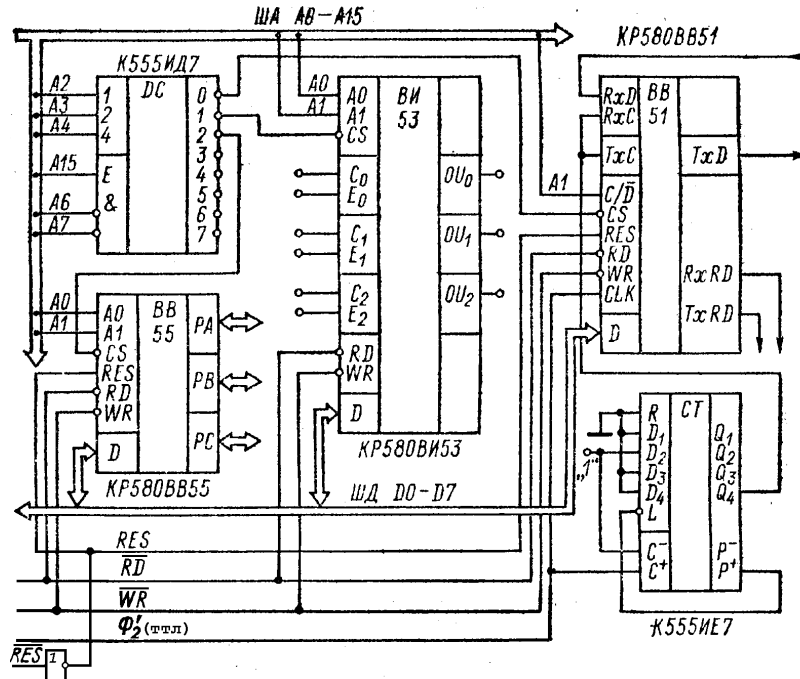


Рис. 5. Схема включения устройств ввода-вывода, как ячеек памяти.

Рассмотрим аспекты построения микро-ЭВМ на БИС микропроцессорного комплекта К580.

Типовая схема микро-ЭВМ, включает в себя микропроцессор КР580ВМ80А, генератор КР580ГФ24, формирующий двухфазную последовательность импульсов; системный контроллер КР580ВК28 (КР580ВК38), обеспечивающий буферизацию шины данных микропроцессора и разделение управляющих сигналов чтения/записи для микросхем памяти и устройств ввода-вывода. Буферная схема адреса, построенная на двух микросхемах КР580ВА86, обеспечивает нагрузочную способность по шине адреса.

Необходимый объем памяти (ПЗУ и ОЗУ) обеспечивают микросхемы К537РФ2 (репрограммируемое ПЗУ с УФ стиранием) и К537РУ10 (статическое ЗУ). В состав микро-ЭВМ могут входить периферийные адаптеры КР580ВВ55 (параллельный интерфейс) КР580ВВ51 (интерфейс последовательного ввода - вывода). В функции параллельного периферийного адаптера обычно входит опрос матрицы датчиков клавиатуры и работа с устройствами, работающими в параллельном коде (АЦП, принтер). Адаптер последовательного интерфейса обеспечивает связь с устройствами по последовательному каналу.

В качестве устройства отображения информации может быть применен как алфавитно-цифровой дисплей на электронно-лучевой трубке (обслуживается контроллером К580ВГ75), так и многоразрядный семисегментный индикатор. В последнем случае функции отображения и сканирования матрицы клавиатуры удобно совместить, применив БИС контроллера К580ВГ79.

При необходимости формирования точных временных интервалов и генерации импульсных последовательностей в состав микро-ЭВМ может быть введен интервальный таймер К580ВИ53. Перечисленные специализированные контроллеры составляют микропроцессорный комплект К580. Рассмотрим состав комплекта и функциональные возможности отдельных БИС входящих в него.

Микропроцессорный комплект серии К580.

Комплект микросхем серии К580, выполненных по n-МДП и ТТЛШ технологии, характеризуется архитектурным единством, которое обеспечивается автономностью и функциональной законченностью отдельных микросхем, унификацией их интерфейса, программируемостью микросхем, их логической и электрической совместимостью. Восьмиразрядная организация, фиксированный набор команд, большой набор периферийных микросхем различного назначения, относительно высокое быстродействие, умеренное потребление мощности обеспечивают МПК широкое применение при создании средств вычислительной техники: устройств локальной автоматики, контроллеров измерительных приборов и периферийных устройств, микро-ЭВМ для управления технологическими процессами и измерительными системами и др.

Отечественная промышленность (НПО "Кристалл") выпускала микропроцессорный комплект К580 в следующем составе:

- К580ВМ80 - процессор, совместимый с i8080;
- К580ГФ24 – генератор тактовых сигналов;
- К580ВК28(38) - системный контроллер;
- К580ВВ55А - адаптер параллельного интерфейса;
- К580ВВ51 - адаптер последовательного интерфейса;
- К580ВИ53 - интервальный таймер;
- К580ВТ57 - контроллер прямого доступа в память;
- К580ВН59А - контроллер прерываний;
- К580ВГ75 - контроллер электронно-лучевой трубки;
- К580ВГ79 - контроллер клавиатуры и индикации;
- К580ВК91А - контроллер канала общего пользования;
- К580ВА93 – программируемый приемопередатчик.

БИС программируемого параллельного периферийного адаптера КР580ВВ55А предназначена для организации ввода/вывода параллельной информации различного формата и позволяет реализовать большинство известных протоколов обмена по параллельным каналам. БИС параллельного программируемого адаптера (ППА) обычно используется для сопряжения микропроцессорной системы со стандартным периферийным оборудованием (цифровым дисплеем, принтером, телетайпом, различными внешними накопителями) через 8-разрядный двунаправленный канал данных.

БИС последовательного интерфейса KP580VB51 представляет собой универсальный синхронно-асинхронный приемопередатчик (УСАПП) и предназначена для организации обмена между микропроцессором (МП) и внешними устройствами (ВУ) в последовательном формате. УСАПП может принимать данные с 8-разрядной шины данных МП и передавать их в последовательном формате периферийным устройствам, а также получать последовательные данные от периферии и преобразовывать их в параллельную форму для передачи в МП. Обмен данными производится в асинхронном режиме со скоростью передачи до 9.6К бит/с или в синхронном - со скоростью до 56К бит/с. Длина передаваемых символов составляет от 5 до 8 бит. При передаче в МП символов длиной менее 8 бит неиспользуемые биты заполняются нулями. Формат символа включает также служебные биты и необязательный бит контроля по четности.

БИС программируемого интервального таймера KP580BI53 – трёхканальное программируемое устройство, предназначенное для организации работы микропроцессорных систем в режиме реального времени. Микросхема формирует сигналы с различными временными параметрами, задаваемыми программно.

Микросхема KP580BT57 представляет собой БИС четырехканального программируемого контроллера прямого доступа к памяти (ПДП). Каждый из четырех каналов адресует область внешней памяти путем инкрементирования выбранного адреса. Контроллер ПДП имеет приоритетную логику, реализующую запросы от четырех периферийных устройств и производит счет циклов прямого доступа к памяти каждого канала. Контроллер предназначен для высокоскоростного обмена данными между памятью системы и периферийными устройствами путем генерации массива последовательных адресов памяти по требованию периферийного устройства. Микросхема осуществляет двусторонний обмен данными между памятью и периферийным устройством путем формирования в адресном канале микропроцессорной системы параметров заданного массива памяти (до 16 К) и управляющих сигналов без участия микропроцессора.

Для обработки событий, происходящих асинхронно по отношению к выполнению программы, лучше всего подходит механизм прерываний. Прерывание можно рассматривать как некоторое особое событие в системе, требующее моментальной реакции. Использование прерываний при работе с медленными внешними устройствами позволяет совместить ввод/вывод с обработкой данных в центральном процессоре и в результате повышает общую производительность системы.

Для организации режима прерываний в микропроцессорном комплекте KP580 предусмотрена **микросхема KP580BH59 - программируемый контроллер прерываний (ПКП)**, обслуживающий до 8-ми запросов на прерывание микропроцессора, с возможностью программного маскирования. Микросхемы KP580BM59 можно использовать в схеме каскадного включения и тогда число запросов на прерывание, поступающих от внешних устройств, может быть расширено до 64; при включении до 9 микросхем. Микросхема позволяет сократить средства программного обеспечения и реальные затраты времени при выполнении прерываний в системах с приоритетами многих уровней. Приоритеты, закрепленные за внешними устройствами, могут быть изменены в процессе выполнения программы.

Микросхема KP580BГ75 - БИС однокристалльного программируемого контроллера электронно-лучевой трубки (ЭЛТ), предназначена для работы алфавитно-цифровыми дисплеями, в также в видеотерминалах микро-ЭВМ. Основное назначение микросхемы – вывод информации из памяти микро-ЭВМ на экран ЭЛТ, промежуточное хранение информации, управление синхронизацией, параметрами развёрток и изображения. К вспомогательным функциям БИС относятся управление курсором, редактирование строки текста, генерация графических символов.

Программируемый контроллер клавиатуры и матричного дисплея K580BV79 – многофункциональное, многорежимное устройство, основным назначением которого является анализ состояния матрицы двухпозиционных датчиков, сигнализация об изменении их состояния и выдача закодированных номеров сработавших датчиков. Контроллер также можно использовать в качестве стробируемого порта ввода, 8-уровневого регистра с организацией FIFO (первый записан, первый прочитан) и дополнительного набора регистров – ОЗУ с организацией 16 слов x 8 разрядов. Содержимое регистрового ОЗУ может циклически выдаваться через порт вывода синхронно с выдачей управляющих сигналов для обслуживания индикаторных дисплеев. Контроллер рассчитан на работу под программным управлением микропроцессора, но может работать и автономно с аппаратной инициализацией после включения питания в нужный режим работы.

Микросхема KP580BK91A – микропроцессорно управляемое устройство, предназначенное для сопряжения микропроцессоров и однокристалльных ЭВМ с линией коллективного пользования (ЛКП) информационно-измерительной системы. Микросхема осуществляет связь между ЛКП и устройствами, управляемыми микропроцессором. В её функции входит передача данных, протокол синхронизации обмена, процедура адресации приемников/передатчиков, очистка и запуск устройств, запрос обслуживания, параллельный и последовательный опросы.

Микросхема KP580BA93 – программируемый приемопередатчик, предназначенный для использования в устройствах вычислительной техники и измерительной аппаратуре с цифровой обработкой информации.

Микросхема KP580BK28(38) – системный контроллер и буферный регистр данных, применяется в микропроцессорных системах на базе микропроцессора K580BM80 для формирования управляющих сигналов и как буферный регистр данных.

Более подробно состав микропроцессорного комплекта K580 и вопросы, связанные с построением микропроцессорных систем на его основе изучаются в курсе учебной дисциплины “Микропроцессорные системы” (IV курс).

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА KP580BM80

Ознакомимся с возможностями микропроцессора KP580BM80 (i8080) в отношении выполнения различных арифметических, логических и прочих операций. Приводимый материал не претендует на справочную полноту изложения. Более подробное описание системы команд и принципов программирования микропроцессоров можно найти в различной специальной и справочной литературе.

Методы адресации. Код команды состоит из кода операции (КО) и сведений об операндах. Операнды в команде могут быть обозначены по-разному, в зависимости от чего и говорят о разных методах адресации. Во-первых, предусмотрено довольно много однобайтных команд, в которых сам код операции несет информацию о внутренних регистрах МП, куда записаны операнды. Это так называемая регистровая адресация. При регистровой адресации часть байта КО представляет собой код регистра. Регистрам *A, B, C, D, E, H, L* (рис. 1.) при этом соответствуют шестнадцатиричные одноразрядные коды 7, 0, 1, 2, 3, 4, 5.

В двухбайтных командах второй байт может представлять собой непосредственно код операнда. Так реализуется второй метод адресации — непосредственная адресация.

Третий метод адресации — прямая адресация. В этом случае второй и третий байты трехбайтной команды представляют собой адрес ячейки памяти, в которой находится операнд. Команды с прямой адресацией используются также при работе с устройствами ввода — вывода. Но адрес соответствующего порта ввода или вывода задается одним байтом, так что код такой команды содержит только два байта.

Четвертый метод адресации — косвенная адресация. В этом случае код команды содержит информацию о 16-разрядном регистре, содержащем адрес ячейки памяти, в которой и находится операнд. При косвенной адресации часто в качестве регистровой пары, содержащей адрес, используются регистры *H, L*. При этом в обозначении команды указывается несуществующий регистр *M* (memory), код которого 6. Таким образом, однобайтная команда, код операции которой содержит указание на регистр *M*, означает, что операнд нужно взять из ячейки памяти или поместить в ячейку памяти, адрес которой записан в регистровой паре *HL*.

Наконец, последний, **пятый метод адресации**, используемой в МП 8080 — это стековая адресация. Как уже указывалось, в этом случае однобайтная команда работы со стеком не несет информации об адресе, просто используется очередная ячейка стековой памяти.

Краткое описание команд МП 8080, за исключением команд перехода, дано в табл. 1. В этой таблице приняты следующие обозначения. Буквой *r* обозначается один из регистров микропроцессора (*B, C, D, E, H, L, M* или *A*). Этим регистрам соответствуют коды *R* (0, 1, 2, 3, 4, 6, 6, 7). В двухбайтной команде код операнда обозначен как N_0 . В трехбайтной команде коды второго и третьего байтов N_0N_1 указывают шестнадцатиричный адрес ячейки памяти $N_1N_0 = N_116^1 + N_016^0$. Все приводимые в таблице коды операций также представлены шестнадцатиричными числами. Запись в столбце «Код операции» формулы вида $06+8R$ означает что нужно код *R*, соответствующий реги-

стру r , умножить на 8 и к полученному произведению прибавить 6. Найденное таким путем шестнадцатиричное число и будет являться кодом операции. Стрелки, используемые в столбце «Содержание команды», показывают направления пересылки данных. В частности запись $r \leftarrow N_0$ означает, что число N_0 помещается в регистр r . Двухбайтное число, заключенное в квадратные скобки $[N_1N_0]$, означает, что данные следует взять из ячейки памяти или поместить в ячейку памяти, адрес которой равен N_1N_0 . Однобайтное число в круглых скобках — это адрес (номер) порта ввода или вывода. Остальные обозначения будут расшифрованы ниже при описании отдельных групп команд.

Команды загрузки и пересылки помещены в первых 18 строках табл. 1. Здесь имеются команды со всеми упомянутыми выше видами адресации: регистровой, непосредственной, прямой, косвенной, стековой. Имеются также команды взаимной пересылки, например XCHG. В соответствии с этой командой взаимноменяется местами содержимое регистровых пар DE и HL .

По команде PCHL содержимое регистровой пары HL загружается в программный счетчик PC . Таким образом, по существу, это команда перехода. Действительно, помещение некоторого числа в программный счетчик означает, что в следующем цикле будет выполняться команда, код которой хранится в памяти по адресу, соответствующему этому числу.

Команда SPHL (так же как и команда LXI SP), позволяет разместить стек в конкретной области пространства памяти. По этой команде в указатель стека SP загружается двухбайтное число из регистровой пары HL .

Команды PUSH и POP — это команды пересылки двухбайтных чисел из регистровых пар в стек и обратно. При этом команда PUSH PSW означает загрузку в стек (S) содержимого пары регистров: аккумулятора A и регистра признаков (регистра флагов) F . Пересылка в эти регистры двух байт информации из стека выполняется с помощью команды POP PSW.

Самую многочисленную группу команд пересылки составляют команды вида MOV r_1, r_2 . Поскольку число регистров r_1 и r_2 равно восьми (B, C, D, E, H, L, M, A), то в принципе может быть всего 64 таких команды. Однако на самом деле их 63, так как команда MOV M, M отсутствует. Но команды, подобные MOV B, B , присутствуют. Они не вызывают никаких действий.

Команды логических операций помещены в строках 19-26 табл. 1. При расшифровке этих команд приняты следующие обозначения: \wedge — операция И, \vee — операция ИЛИ, \oplus — операция «исключающее ИЛИ». Перечисленные логические операции выполняются поразрядно. Запись $A \leftarrow A \wedge r$, например, означает, что в каждый разряд регистра-аккумулятора помещается одноразрядное двоичное число, представляющее собой функцию И чисел в аналогичных разрядах прежнего содержимого аккумулятора и содержимого регистра r .

Команда CMP — это команда сравнения чисел, хранящихся в аккумуляторе и регистре r . При этом содержимое аккумулятора не изменяется, но разряды регистра флажков F устанавливаются в соответствии с разностью $r - A$. Подобным же образом осуществляется сравнение числа в аккумуляторе и числа N_0 , приведенного во втором байте команды CPI N_0 .

Команды арифметических операций размещены в строках 27—42 табл. 1. Буква s в описании этих команд означает содержимое разряда переноса регистра флагов F .

Системные и специальные команды — это команды, помещенные в строках 43—53 табл. 1. Команда NOP означает просто пропуск одного цикла и никак не влияет на содержимое регистров (кроме РС) микропроцессорной системы. Команда HLT останавливает работу МП до прихода запроса прерывания или сигнала сброса. Команда CMA инвертирует число, записанное в аккумуляторе. Команды CMC и STC оперируют с содержимым бита переноса регистра флажков. По команде DAA корректируется результат, если операнды двоично-десятичные. Команды IN N_0 и OUT N_0 позволяют произвести обмен информацией с портами ввода — вывода, имеющими адрес N_0 .

Для того чтобы запретить или разрешить микропроцессору реагировать на команды прерывания, используются команды DI и EI. По команде RST n в МП вводится вектор прерывания. Напомним, что если нет запрета на обработку прерываний (команда DI), то по сигналу запроса прерывания INT, МП выставляет нулевой сигнал на выходе INTE. При этом устройство, запросившее прерывание, должно выдать на шину данных код команды. МП считывает эту команду и далее ее выполняет. Обычно здесь используется команда RST n . По этой команде в стеке запоминается содержимое программного счетчика РС, и затем в этот счетчик помещается двухбайтное шестнадцатеричное число $8n$. Таким образом осуществляется переход к работе с той областью памяти, в которой записана команда обработки данного прерывания. Заметим, что команда RST 0 возвращает МП к первой ячейке памяти с адресом 0000H, так что эта команда действует аналогично сигналу сброса RESET.

Команды сдвигов приведены в последних четырех строках табл.1. По этим командам осуществляется кольцевой сдвиг числа в аккумуляторе на один двоичный разряд влево или вправо, с включением или без включения в кольцо разряда переноса из регистра флажков F . При этом символом B_m обозначен код m -го разряда (бита) числа в аккумуляторе (старший разряд — B_7).

Команды переходов представлены в табл. 2. Эти команды делятся на три группы: *простые переходы* (первая буква обозначения команды J), *переходы к подпрограммам* (первая буква C) и *возвраты из подпрограмм* (первая буква R). Переход к подпрограмме отличается от простого перехода тем, что при этом автоматически запоминается в стеке номер команды (показание программного счетчика в момент осуществления перехода). Затем, когда по команде возврата будет осуществляться обратный переход, этот номер будет извлечен из стека и помещен снова в программный счетчик.

Условные переходы, как следует из табл. 2, происходят в зависимости от кода в отдельных разрядах регистра флажков. Например, простой переход JM реализуется в том случае, если в разряде знака этого регистра записана единица ($Z=1$). При этом же условии реализуется переход к подпрограмме по команде CM и возврат из подпрограммы по команде RM.

В табл. 2 для экономии места команды простого перехода и перехода к подпрограмме приведены без указаний на коды второго и третьего байтов. Эти команды трехбайтные (в отличие от однобайтных команд возврата); во втором и третьем байтах содержится информация об адресе, по которому следует осуществлять переход. Этот адрес при переходе и записывается в программный счетчик.

В табл. 1 и 2 коды операций приведены в виде шестнадцатеричных чисел. Это сделано для более компактной записи этих кодов. В МП-системе эти коды, равно как и вся прочая информация, представлены двоичными числами. Для примера найдем двоичный код, соответствующий команде загрузки в регистр L

числа 3FH. Эта команда приведена в строке 1 табл.1. Учитывая, что код регистра L равен пяти, находим шестнадцатиричный код операции: $06+8\cdot5=46D=2EH$. Таким образом, шестнадцатиричный код рассматриваемой двухбайтной команды имеет вид $2E3F$. Соответствующий двоичный машинный код будет представлен числом $00_{10}1110_{10}00111111_2$.

Команды, номера которых в табл. 1 подчеркнуты, оказывают влияние хотя бы на один разряд регистра флагов F . Остальные команды (табл. 1 и 2) не изменяют состояния этого регистра.

Таблица 1

№№ п.п.	Обозначение команды	КОД операции (HEX)	<u>Содержание команды</u>
<u>Команды загрузки и пересылки</u>			
1	MVI r, N_0	06+8R	$r \leftarrow N_0, \quad r=B, C, D, E, H, L, M, A$
2	LXI r, N_0N_1	01+8R	$rp \leftarrow N_0N_1, \quad rp=B, D, H$
3	LXI SP, N_0N_1	31	$SP \leftarrow N_0N_1$
4	LDA N_0N_1	3A	$A \leftarrow [N_0N_1]$
5	STA N_0N_1	32	$[N_0N_1] \leftarrow A$
6	LDAX r	0A+8R	$A \leftarrow [rp], \quad rp=B, D$
7	STAX r	02+8R	$[rp] \leftarrow A, \quad rp=B, D$
8	MOV r_1, r_2	40+8R ₁ +R ₂	$r_1 \leftarrow r_2, \quad r=B, C, D, E, H, L, M, A$
9	LHLD N_0N_1	2A	$L \leftarrow [N_0N_1], \quad H \leftarrow [N_0N_1+1]$
10	SHLD N_0N_1	22	$[N_0N_1] \leftarrow L, \quad [N_0N_1+1] \leftarrow H$
11	XTHL	E3	$HL \leftrightarrow SP$
12	XCHG	EB	$D \leftrightarrow H, \quad E \leftrightarrow L$
13	PCHL	E9	$PC \leftarrow HL$
14	SPHL	F9	$SP \leftarrow HL$
15	PUSH r	C5+8R	$S \leftarrow rp, \quad rp=B, D, H$
16	PUSH PSW	F5	$S \leftarrow AF$
17	POP r	C1+8R	$rp \leftarrow S, \quad rp=B, D, H$
<u>18</u>	POP PSW	F1	$AF \leftarrow S$

Команды логических операций			
<u>19</u>	ANA r	A0+R	$A \leftarrow A \wedge r, \quad r=B, C, D, E, H, L, M, A$
<u>20</u>	ORA r	80+R	$A \leftarrow A \vee r, \quad r=B, C, D, E, H, L, M, A$
<u>21</u>	XRA r	A8+R	$A \leftarrow A \oplus r, \quad r=B, C, D, E, H, L, M, A$
<u>22</u>	ANI N ₀	E6	$A \leftarrow A \wedge N_0$
<u>23</u>	ORI N ₀	F6	$A \leftarrow A \vee N_0$
<u>24</u>	XRI N ₀	EE	$A \leftarrow A \oplus N_0$
<u>25</u>	CMP r	B8+R	$A - r, \quad r=B, C, D, E, H, L, M, A$
<u>26</u>	CPI N ₀	FE	$A - N_0$
Команды арифметических операций			
<u>27</u>	INR r	04+8R	$r \leftarrow r+1, \quad r=B, C, D, E, H, L, M, A$
<u>28</u>	DCR r	05+8R	$r \leftarrow r-1, \quad r=B, C, D, E, H, L, M, A$
29	INX r	03+8R	$rp \leftarrow rp+1, \quad rp=B, D, H$
30	INX SP	33	$SP \leftarrow SP+1$
31	DCX r	0B+8R	$rp \leftarrow rp-1, \quad rp=B, D, H$
32	DCX SP	3B	$SP \leftarrow SP-1$
<u>33</u>	ADD r	80+R	$A \leftarrow A+r, \quad r=B, C, D, E, H, L, M, A$
<u>34</u>	SUB r	90+R	$A \leftarrow A-r, \quad r=B, C, D, E, H, L, M, A$
<u>35</u>	ADC r	88+R	$A \leftarrow A+r+c, \quad r=B, C, D, E, H, L, M, A$
<u>36</u>	SBB r	98+R	$A \leftarrow A-r-c, \quad r=B, C, D, E, H, L, M, A$
<u>37</u>	ADI N ₀	C6	$A \leftarrow A+N_0$
<u>38</u>	SUI N ₀	D6	$A \leftarrow A-N_0$
<u>39</u>	ACI N ₀	CE	$A \leftarrow A+N_0+c$
<u>40</u>	SBI N ₀	DE	$A \leftarrow A-N_0-c$
<u>41</u>	DAD r	09+8R	$HL \leftarrow HL+rp, \quad rp=B, D, H$
<u>42</u>	DAD SP	39	$HL \leftarrow HL+SP$

Системные и специальные команды			
43	NOP	00	Пустая операция (Нет операции)
44	HLT	76	Останов
45	CMA	2F	$A \leftarrow \neg A$
<u>46</u>	CMC	3F	$c \leftarrow \neg c$
<u>47</u>	STC	37	$c \leftarrow 1$
<u>48</u>	DAA	27	Десятичная коррекция
49	IN N_0	DB	$A \leftarrow (N_0)$
50	OUT N_0	D3	$(N_0) \leftarrow A$
51	DI	F3	Запрет прерывания
52	EI	F8	Разрешение прерывания
53	RST n	$C7+8n$	$S \leftarrow PC, PC \leftarrow 8n, n=0,1,2,3,4,5,6,7$
Команды сдвигов			
<u>54</u>	RLC	07	$B_{m+1} \leftarrow B_m, B_0 \leftarrow B_7, c \leftarrow B_7$
<u>55</u>	RRC	0F	$B_m \leftarrow B_{m+1}, B_7 \leftarrow B_0, c \leftarrow B_0$
<u>56</u>	RAL	17	$B_{m+1} \leftarrow B_m, B_0 \leftarrow c, c \leftarrow B_7$
<u>57</u>	RAR	1F	$B_m \leftarrow B_{m+1}, B_7 \leftarrow c, c \leftarrow B_0$

Таблица 2

Команды переходов			
№№ п.п.	Обозначение команды	Коды операции (HEX)	<u>Условие перехода</u>
1	JMP, CALL, RET	C3, CD, C9	Без условий
2	JC, CC, RC	DA, DC, D8	c = 1
3	JNC, CNC, RNC	D2, D4, D0	c = 0
4	JZ, CZ, RZ	CA, CC, C8	Z = 1
5	JNZ, CNZ, RNZ	C2, C4, C0	Z = 0
6	JP, CP, RP	F2, F4, F0	S = 0
7	JM, CM, RM	FA, FC, F8	S = 1
8	JPE, CPE, RPE	EA, EC, E8	P = 1
9	JPO, CPO, RPO	E2, E4, E0	P = 0

Программирование МП-системы заключается в составлении программы и последующем помещении кодов команд в ячейки памяти системы. Программа, естественно, может непосредственно составляться в виде последовательности двоичных машинных кодов команд. Однако такая программа совершенно ненаглядна и ее написание представляет собой трудную и чреватую ошибками работу. Более удобно пользоваться при написании программы mnemonic обозначениями команд, приведенными в табл. 1 и 2. Существует специальная программа — ассемблер, с помощью которой можно на ЭВМ перевести написанную таким образом программу в машинные коды. Соответственно программа, написанная с использованием mnemonic обозначений команд, — это программа на языке ассемблера.

Весьма удобно составлять программу, подлежащую реализации в МП-системе, на языке программирования высокого уровня (си, паскаль, бейсик и т.п.). Затем такая программа с помощью ЭВМ тоже переводится в последовательность машинных кодов с помощью специальной программы — транслятора. Однако получаемая при этом конечная программа оказывается, как правило, заметно длиннее программы, составленной с помощью языка *ассемблера*. Это означает, что для хранения этой программы потребуется больший объем памяти. Так что выбор метода программирования МП-системы может в конечном счете зависеть от наличия и стоимости запоминающих устройств и от резерва времени для выполнения в системе более длинной программы.

Более подробно методы и правила программирования рассматриваются в курсе учебной дисциплины “Микропроцессорные системы”.